Accelerated and interpretable oblique random survival forests

Abstract

The oblique random survival forest (RSF) is an ensemble supervised learning method for right-censored outcomes. Trees in the oblique RSF are grown using linear combinations of predictors, whereas in the standard RSF, a single predictor is used. Oblique RSF ensembles have high prediction accuracy, but assessing many linear combinations of predictors induces high computational overhead. In addition, few methods have been developed for estimation of variable importance (VI) with oblique RSFs. We introduce a method to increase computational efficiency of the oblique RSF and a method to estimate VI with the oblique RSF. Our computational approach uses Newton-Raphson scoring in each non-leaf node, We estimate VI by negating each coefficient used for a given predictor in linear combinations, and then computing the reduction in out-of-bag accuracy. In benchmarking experiments, we find our implementation of the oblique RSF is hundreds of times faster, with equivalent prediction accuracy, compared to existing software for oblique RSFs. We find in simulation studies that 'negation VI' discriminates between relevant and irrelevant numeric predictors more accurately than permutation VI, Shapley VI, and a technique to measure VI using analysis of variance. All oblique RSF methods in the current study are available in the `aorsf` R package.

Keywords: Supervised learning, Computational efficiency, Variable importance

1 Introduction

Risk prediction may reduce the burden of disease by guiding strategies for prevention and treatment in a wide range of domains (Moons et al., 2012). The random survival forest (RSF; Ishwaran et al. (2008); Hothorn et al. (2006)) is a supervised learning algorithm that has been used frequently for risk prediction (Wang and Li, 2017). Similar to random forests (RFs) for classification and regression (Breiman, 2001), The RSF is a large set of de-correlated and randomized decision trees, with each tree contributing to the ensemble's prediction function. Notable characteristics of the RSF include uniform convergence of its ensemble survival prediction function to the true survival function, first shown by Ishwaran and Kogalur (2010) and later by Cui et al. (2017) under more general conditions. However, Cui et al. (2017) noted that the RSF is at a disadvantage when predictors are correlated and some are not relevant to the censored outcome, which is a strong possibility when large clinical and 'omic' databases are leveraged for risk prediction.

A potential approach to improve the RSF when predictors are correlated and some are not relevant to the censored outcome is to use oblique trees instead of axis based trees. Axis based trees split data using a single predictor, creating decision boundaries that are perpendicular or parallel to axes of the predictor space (see Breiman et al., 2017, Chapter 2). Oblique trees split data using a linear combination of predictors, creating decision boundaries that are neither parallel nor perpendicular to axes of their contributing predictors (see Breiman et al., 2017, Chapter 5). Oblique trees may create more adequate partitions of a predictor space compared to axis-based trees, as demonstrated in Figure **??**. Menze et al. (2011) examined prediction accuracy of RFs in the presence of correlated predictors and found that oblique RFs had substantially higher prediction accuracy compared to axis-based RFs. Similarly, Jaeger et al. (2019) found that growing RSFs with oblique rather than axis-based trees reduced the RSF's concordance error, with improvements ranging from 2.5% to 24.9% depending on the data analyzed.

Despite the potential for higher accuracy, oblique trees have at least two notable drawbacks compared to axis-based trees. First, finding a locally optimal oblique decision rule may require exponentially more computation than an axis-based rule. If p predictors are potentially used to split n observations, up to $\mathcal{O}(n^p)$ oblique splits can be assessed versus $\mathcal{O}(n \cdot p)$ axis-based splits (Heath et al., 1993; Murthy et al., 1994). Second, although variable importance (VI) is one of the

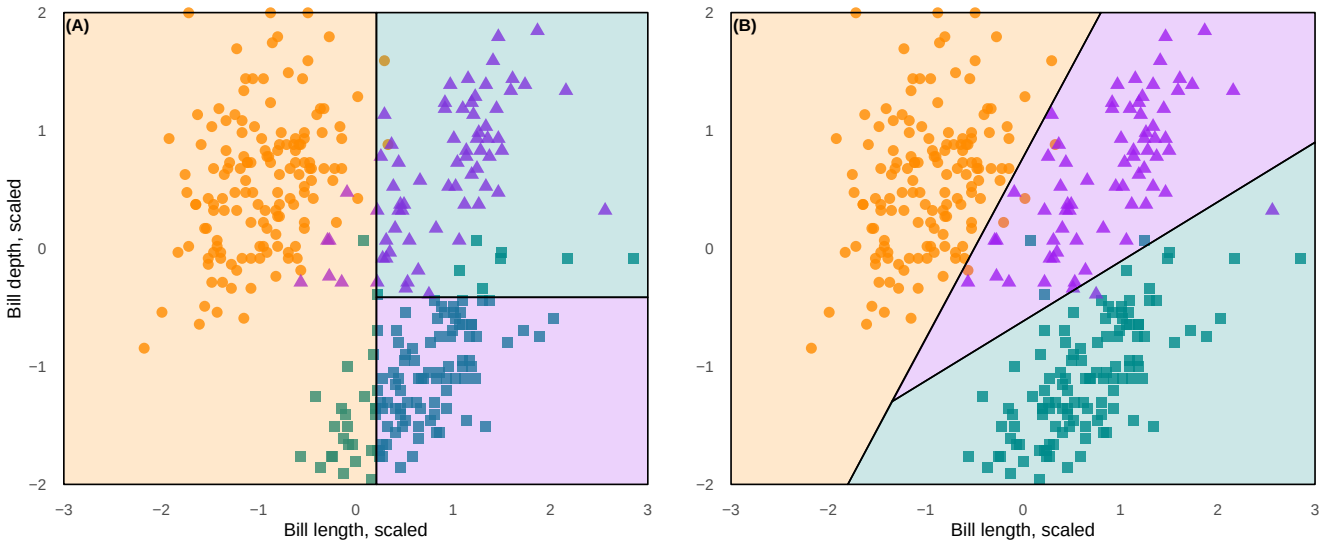

Figure 1: Decision boundaries from an axis based (panel A) and oblique (panel B) decision tree used to classify penguin species based on bill depth and bill length. The decision boundary from the oblique tree is better able to capture the geometry of this data, leading to fewer mis-classified penguins.

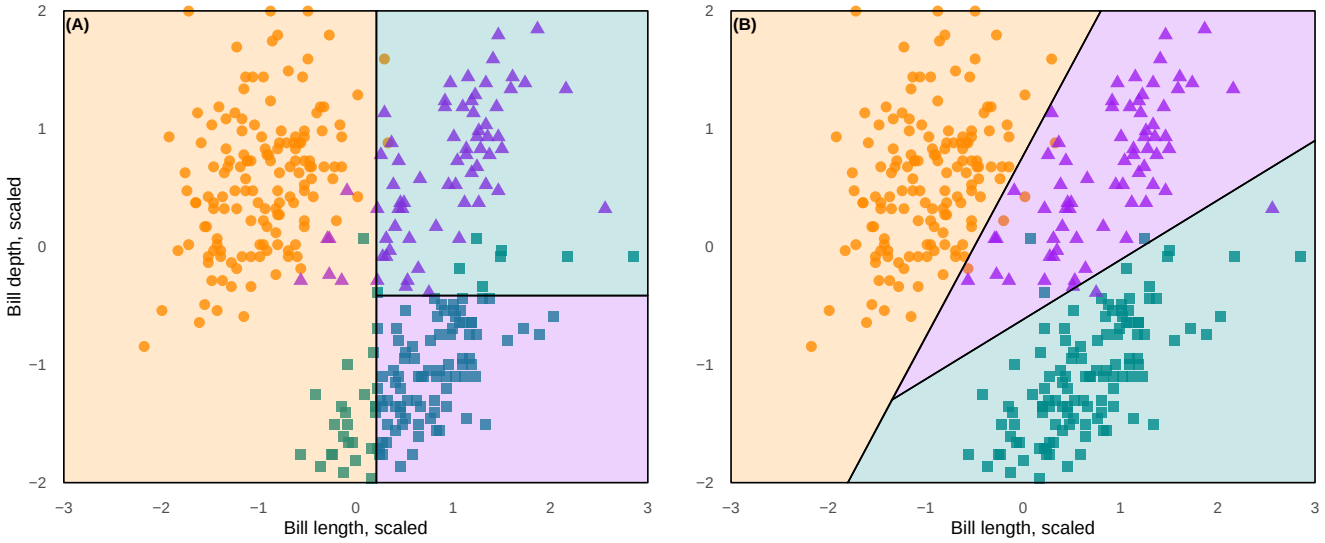

Figure 2: Decision boundaries from an axis based (panel A) and oblique (panel B) decision tree used to classify penguin species based on bill depth and bill length. The decision boundary from the oblique tree is better able to capture the geometry of this data, leading to fewer mis-classified penguins.

most widely used strategies to interpret RFs (Ishwaran and Lu, 2019), few studies have investigated VI for oblique RFs (see Menze et al., 2011, Section 5), and fewer have investigated VI specifically for the oblique RSF. Without general methodology to estimate VI, interpretation of oblique RFs is challenging.

The aim of this paper is to introduce methodology that improves the computational efficiency and interpretation of oblique RSFs. Section 2 reviews prior work, introduces our method to reduce the computational cost of oblique RSFs (*i.e.*, accelerate them), and introduces 'negation VI', a method to estimate VI with oblique RSFs that does not require permutation of data. We describe benchmarking experiments and simulation studies to evaluate these methods in Section 3, and present results in Section 4. In Section 5, we summarize results from the current study, connecting our findings to prior work and outlining potential future research topics. All oblique RSF methods introduced in the current study are available in the `aorsf` R package (Jaeger et al., 2022).

2 Methods and materials

Sections 2.1 and 2.2 briefly summarize prior studies that have developed methods related to the oblique RSF and VI, respectively. Section 2.3 describes our approach to reduce computational overhead of the oblique RSF and Section 2.4 introduces negation VI, a novel technique to estimate VI in oblique RFs.

2.1 Axis-based and oblique random forests

After Breiman (2001) introduced the axis-based and oblique RF, numerous methods were developed to grow oblique RFs for classification or regression tasks (Menze et al., 2011; Zhang and Suganthan, 2014; Rainforth and Wood, 2015; Zhu et al., 2015; Poona et al., 2016; Qiu et al., 2017; Tomita et al., 2020; Katuwal et al., 2020). However, oblique splitting approaches for classification or regression may not generalize to censored outcomes (*e.g.*, see Zhu, 2013, Section 4.5.1), and most research involving the RSF has focused on forests with axis-based trees (Wang and Li, 2017).

Hothorn et al. (2006) developed an axis-based RSF in their framework for unbiased recursive partitioning, more commonly referred to as the conditional inference forest (CIF). Zhou et al. (2016) developed a rotation forest based on the CIF and Wang and Zhou (2017) developed a method for

extending the predictor space of the CIF. Ishwaran et al. (2008) developed an axis-based RSF with strict adherence to the rules for growing trees proposed in Breiman (2001). Jaeger et al. (2019) developed the oblique RSF following the bootstrapping approach described in Breiman's original RF and incorporating early stopping rules from the CIF.

Fast algorithms to fit axis-based RSFs are available in the `randomForestSRC` R package (Ishwaran and Kogalur, 2019) and the `ranger` (Wright and Ziegler, 2017) R package. `randomForestSRC` provides a unified interface to grow RFs in a wide range of analyses, and `ranger` is designed to grow RFs efficiently using high dimensional data. Fast algorithms to fit the CIF are provided by the `party` R package (Hothorn et al., 2010), which provides a computational toolbox for recursive partitioning using conditional inference trees. Jaeger et al. (2019) developed the `obliqueRSF` package and found it was approximately 30 times slower than `party` and nearly 200 times slower than `randomForestSRC`. Few studies have developed software with fast algorithms for oblique RSFs that have comparable speed compared to algorithms for axis-based RSFs.

2.2 Variable importance

Several techniques to estimate VI have been developed since Breiman (2001) introduced permutation VI, which is defined for each predictor as the difference in a RF's estimated prediction error before versus after the predictor's values are randomly permuted. Strobl et al. (2007) identified bias in permutation VI driven by categorical predictors and bootstrap sampling, and proposed a permutation VI measure based on unbiased recursive partitioning (Hothorn et al., 2006). Menze et al. (2011) introduced an approach to estimate VI for oblique RFs that computes an analysis of variance (ANOVA) table in non-leaf nodes to obtain p-values for each predictor contributing to the node. The ANOVA VI[1] is then defined for each predictor as the number of times a p-value associated with the predictor is ≤ 0.01 while growing a forest. Lundberg and Lee (2017) introduced a method to estimate VI using SHapley Additive exPlanation (SHAP) values, which estimates the contribution of a predictor to a model's prediction for a given observation. SHAP VI is computed for each predictor by taking the mean absolute value of SHAP values for that predictor across all observations in a given set. With the exception of Menze et al. (2011), few studies have evaluated estimation of VI

[1]Menze et al. (2011) name their method 'oblique RF VI', but we use the name 'ANOVA VI' in this article to avoid confusing Menze's approach with other approaches to estimate VI for oblique RFs.

using oblique RFs, and fewer have examined VI specifically for the oblique RSF.

2.3 The accelerated oblique random survival forest

Consider the usual framework for right-censored time-to-event outcomes with training data

$$\mathcal{D}_{\text{train}} = \{(T_i, \delta_i, \boldsymbol{x}_i)\}_{i=1}^{N_{\text{train}}}.$$

Here, T_i is the event time if $\delta_i = 1$ or the censoring time if $\delta_i = 0$, and $\boldsymbol{x}_i$ is a vector of predictors values. Assuming there are no ties, let $t_1 < \ldots < t_m$ denote the m unique event times in $\mathcal{D}_{\text{train}}$.

To accelerate the oblique RSF, we propose to identify linear combinations of predictor variables in non-leaf nodes by applying Newton Raphson scoring to the partial likelihood function of the Cox regression model:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{m} \frac{e^{\boldsymbol{x}_{j(i)}^T \boldsymbol{\beta}}}{\sum_{j \in R_i} e^{\boldsymbol{x}_j^T \boldsymbol{\beta}}}, \tag{1}$$

where R_i is the set of indices, j, with $T_j \geq t_i$ (*i.e.*, those still at risk at time t_i), and $j(i)$ is the index of the observation for which an event occurred at time t_i. Newton Raphson scoring is an exceptionally fast estimation procedure, and the `survival` package (Therneau, 2022b) includes documentation that outlines how to efficiently program it (Therneau, 2022a). Briefly, a vector of length `mtry` (*i.e.*, the number of randomly selected predictors) with estimated regression coefficients, $\hat{\beta}$, is updated in each step of the procedure based on its first derivative, $U(\hat{\beta})$, and second derivative, $H(\hat{\beta})$:

$$\hat{\beta}^{k+1} = \hat{\beta}^k + U(\hat{\beta} = \hat{\beta}^k)\, H^{-1}(\hat{\beta} = \hat{\beta}^k)$$

For statistical inference, it is recommended to continue updating $\hat{\beta}$ by completing additional iterations of Newton Raphson scoring until a convergence threshold is met. However, since an estimate of $\hat{\beta}$ is created by the first iteration of Newton Raphson scoring, only one iteration of Newton Raphson scoring is needed to identify a valid linear combination of predictors. Moreover, computing U and H requires computation and exponentiation of the vector $\boldsymbol{x}\hat{\beta}$, but these steps can be skipped on the first iteration of Newton Raphson scoring if an initial value of $\hat{\beta} = 0$ is chosen, allowing for a reduction in computing operations and removing the need to scale predictor

values prior to initiating the Newton Raphson algorithm.[2] In Section 4.1, we formally test whether growing oblique survival trees using one iteration of Newton Raphson scoring provides equivalent prediction accuracy compared to trees where iterations are completed until a convergence threshold is met.

Algorithm 1 presents our approach to fitting an oblique survival tree in the accelerated oblique RSF using default values from the `aorsf` R package. Several steps are taken to reduce computational overhead. First, memory is conserved by conducting bootstrap resampling via random integer-valued weights, rather than using a bootstrapped copy of the original data. Memory conservation also takes place in terminal nodes, where we restrict estimation of the survival and cumulative hazard function to event times that occur among observations in the node. Second, early stopping is applied to the tree-growing procedure if a statistical criterion is not met. In our case, the criterion is based on the magnitude of a log-rank test statistic corresponding to splitting the data at a current node. Third, instead of greedy recursive partitioning, we use 'good enough' partitioning. More specifically, instead of computing a log-rank test statistic for several different linear combinations of variables and proceeding with the highest scoring option, we identify an optimal cut-point for one linear combination of variables and assess whether using this combination will create a split that passes the criterion for splitting a node. If it does not pass the criterion, then another linear combination will be tested, with the maximum number of attempts set by the parameter `n_retry`. Often a 'good-enough' split can be found in just one attempt when the training set is large, which gives the accelerated oblique RSF a computational advantage in larger training sets compared to greedy partitioning.

2.4 Negation variable importance

This Section introduces negation VI, which is similar to permutation VI in that it measures how much a model's prediction error increases when a variable's role in the model is de-stabilized. Specifically, negation VI measures the increase in an oblique RF's prediction error after flipping the sign of all coefficients linked to a variable (*i.e.*, negating them). As negating a coefficient effectively flips decision boundaries around the corresponding predictor's axis, scaling numeric predictors to

[2]Predictors are scaled prior to initiating the Newton Raphson algorithm to avoid exponentiation of large numbers. However, if only one iteration is completed with an initial value of 0 for $\hat{\beta}$, then $\exp(\boldsymbol{x}\hat{\beta}) = 1$.

Algorithm 1 Accelerated oblique random survival tree using default parameters.

Require: Training data $\mathcal{D}_{\text{train}} = \{(T_i, \delta_i, \boldsymbol{x}_i)\}_{i=1}^{N_{\text{train}}}$, mtry $= \sqrt{\text{ncol}(\boldsymbol{x}_{\text{train}})}$, n_split $= 5$, n_retry $= 3$, and split_min_stat $= 3.841459$

$\mathcal{T} \leftarrow \emptyset$
$w \leftarrow$ sample(from $= \{0, \ldots, 10\}$, size $=$ nrow($\boldsymbol{x}_{\text{train}}$), replace $=$ T)
$\mathcal{D}_{\text{in-bag}} \leftarrow$ subset($\mathcal{D}_{\text{train}}$, rows $=$ which($w > 0$))
$w \leftarrow$ subset(w, which($w > 0$))
node_assignments $\leftarrow$ rep(1, times $=$ nrow($\boldsymbol{x}_{\text{in-bag}}$))
nodes_to_split $\leftarrow \{1\}$
while nodes_to_split $\neq \emptyset$ **do**
    for node $\in$ nodes_to_split **do**
        n_try $\leftarrow 1$
        node_rows $\leftarrow$ which(node_assignments $\equiv$ node)
        node_cols $\leftarrow$ sample(from $= \{1, \ldots, \text{ncol}(\boldsymbol{x})\}$, size $=$ mtry, replace $=$ F)
        $\mathcal{D}_{\text{node}} \leftarrow$ subset($\mathcal{D}_{\text{in-bag}}$, rows $=$ node_rows, columns $=$ node_cols)
        $\beta \leftarrow$ newt_raph($\mathcal{D}_{\text{node}}$, weights $=$ subset(w, node_rows), max_iter $= 1$)
        $\eta \leftarrow \boldsymbol{x}_{\text{node}} \times \beta$
        $\mathcal{C} \leftarrow$ sample(from $=$ unique(η), size $=$ n_split, replace $=$ F)
        $c \leftarrow \text{argmax}_{c^* \in \mathcal{C}} \{\text{log\_rank\_stat}(\eta, c^*)\}$
        if log_rank_stat(η, c) $\geq$ split_min_stat **then**
            $\mathcal{T} \leftarrow$ add_node($\mathcal{T}$, name $=$ node, beta $= \beta$, cutpoint $= c$)
            ▷ Right node logic omitted for brevity (identical to left node logic)
            node_left_name $\leftarrow$ max(node_assignments) $+ 1$
            node_left_rows $\leftarrow$ subset(node_rows, which($\eta \leq c$))
            subset(node_assignments, node_left_rows) $\leftarrow$ node_left_name
            if is_splittable(subset(node_assignments, node_left_rows)) **then**
                nodes_to_split $\leftarrow$ nodes_to_split $\cup$ node_left_name
            else
                $\mathcal{T} \leftarrow$ add_leaf($\mathcal{T}$, data $=$ subset($\mathcal{D}_{\text{node}}$, rows $=$ node_left_rows))
            end if
        else if n_try $\leq$ n_retry **then**
            n_try $\leftarrow$ n_try $+ 1$
            go to 11
        else
            $\mathcal{T} \leftarrow$ add_leaf($\mathcal{T}$, data $= \mathcal{D}_{\text{node}}$)
        end if
        nodes_to_split $\leftarrow$ nodes_to_split $\setminus$ {node}
    end for
end while
return $\mathcal{T}$

have a mean of zero and standard deviation of one is recommended.[3] For the current study, we use Harrell's concordance (C)-statistic (Harrell et al., 1982) to measure change in prediction error when computing negation VI.

To demonstrate negation VI, consider a classification task where the goal is prediction of penguin species (chinstrap, gentoo, or adelie) based on bill length and bill depth (Horst et al., 2020). Scaling these predictors to be centered at 0, we find oblique decision boundaries defined by linear combinations of bill length and bill depth correctly classify most of the data (Figure 3, Panel A). Permuting the values of bill length leads to several mis-classified observations, suggesting that bill length is an important predictor (Figure 3, Panel B). However, inspecting the permuted data shows that a number of observations moved to a region in the predictor space where there were previously no observations. Moving data to unobserved or perhaps unobservable regions of the predictor space may cause extrapolation error, which Hooker et al. (2021) identified as a cause of bias in permutation importance. Negating the coefficients for bill length in the linear combinations that define our decision boundaries causes the boundaries to flip across bill length's axis (Figure 3, Panel C). This leads to several mis-classified observations, suggesting that bill length is an important predictor without distorting the joint distribution of bill length and bill depth.

Negation VI is an extension of "anti VI", a VI technique for axis-based trees which became the default VI method for `randomForestSRC` in version 2.14.0. Anti VI reverses the direction of all decision nodes that use a specific variable, and then reassesses the ensemble prediction error. So, if an axis-based decision rule were defined as $x > 5 \Rightarrow$ send data to right node, the decision rule would become $x > 5 \Rightarrow$ send data to left node when computing the importance of a predictor x. Put in a way that makes the connection to negation VI more explicit, the 'noised up' decision rule can be written as $-x > 5 \Rightarrow$ send data to right node.

Negation VI has several strengths. First, it generalizes to any oblique RF (*i.e.*, not just RSFs) using any valid error function, making it both general and flexible.[4] Second, negation is non-random, making it reproducible without setting a random seed and making it slightly faster than permutation importance. Fourth, since negation VI does not permute variables, the analyst need not worry about impossible combinations of predictors that may occur when one predictor is randomly permuted,

[3]The `aorsf` package automatically scales numeric inputs to a mean of zero and standard deviation of one.

[4]The `aorsf` package enables customized functions to be applied in lieu of the default C-statistic.

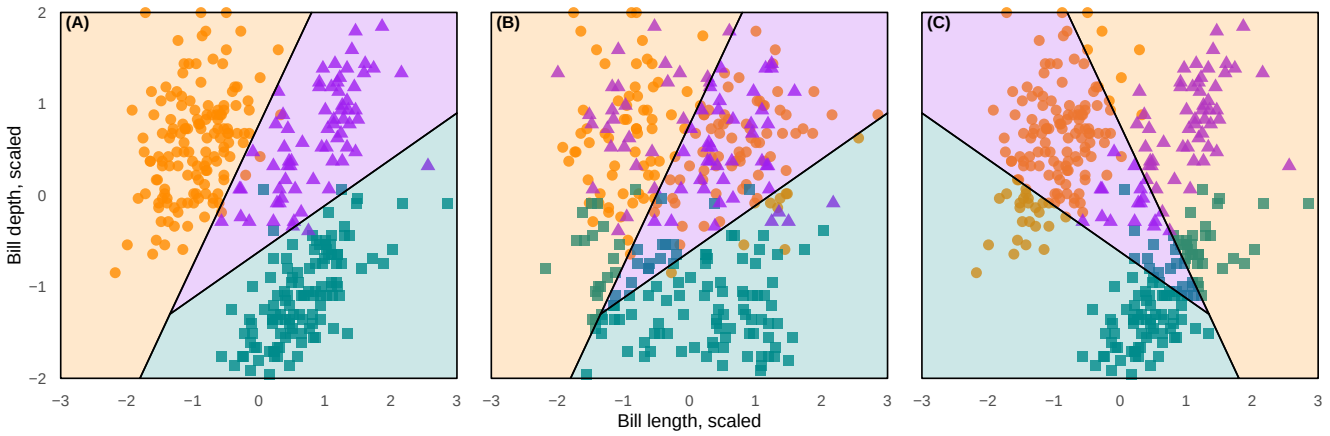

Figure 3: Demonstration of negation and permutation importance for a single oblique tree. Panel A shows the original data and decision boundaries. Panel B shows the data with permuted bill length values. Panel C shows the decision boundaries after negating coefficients for bill length. Permutation and negation both show that bill length is an important predictor, but permuting bill length distorts its joint distribution with bill depth.

such as having a negative status for type 2 diabetes and having Hemoglobin A1c level $\geq 6.5\%$ (a value indicative of type 2 diabetes) as a result of randomly permuting the values of Hemoglobin A1c. However, in scenarios where decision boundaries have symmetry around the origin of the predictor space (*e.g.*, all positive cases lie in a circle centered at the origin, with negative cases sitting outside the circle), negation importance will be less effective than permutation.

3 Numeric experiments

Sections 3.1 and 3.2 present the design of numerical experiments examining the accelerated oblique RSF and negation VI, respectively. Section 3.3 summarizes our approach to evaluating computational efficiency of learning algorithms, with a focus on the accelerated oblique RSF and other RSF implementations. Section 3.4 provides details on computation and code.

3.1 Benchmark of prediction accuracy

The aim of this numeric experiment is to evaluate the prediction accuracy of the accelerated oblique RSF compared to its predecessor (the oblique RSF from the `obliqueRSF` R package) and to several

other machine learning algorithms. Inferences drawn from this experiment include equivalence and inferiority tests based on Bayesian linear mixed models.

3.1.1 Learners

We consider four classes of learners: RSFs (both axis-based and oblique), boosting ensembles, regression models, and neural networks. Specific learners from each class are summarized in Table 1. To facilitate fair comparisons, tuning parameters were harmonized within each class. For example, for RSF learners, we set the minimum node size (a parameter shared by all RSF learners) as 10. Additionally, for RSF learners, the number of randomly selected predictors was the square root of the total number of predictors rounded to the nearest integer, and the number of trees in the ensemble was 500 (a common default value for the number of trees). For boosting, regression, and neural network learners, nested cross-validation was applied to tune relevant model parameters. Specifically, tuning for boosting models included identifying the number of steps to complete. For regression models, tuning was used to identify the magnitude of penalization. For neural networks, the number and density of layers was tuned.

Learner Class	Software	Learners	Description
Random Survival Forests			

Axis based	`RandomForestSRC`
	`ranger`
	`party`
	`rotsf`
	`rsfse`
		`rsf-standard`
		`rsf-extratrees`
		`cif-standard`
		`cif-rotate`
		`cif-spacextend`
			`rsf-standard` grows survival trees following Leo Breiman's random forest algorithm with cut-points selected to maximize a log-rank statistic. `rsf-extratrees` grows survival trees with randomly selected predictors and cut-points. `cif-standard` uses conditional inference. `cif-rotate` extends `cif-standard` by applying principal component analysis to random subsets of data prior to growing each survival tree. `cif-spacextend` derives new predictors for each tree in the ensemble, separately.

Oblique	`obliqueRSF`
	`aorsf`
		`obliqueRSF-net`
		`aorsf-fast`
		`aorsf-cph`
		`aorsf-extratrees`
			Oblique survival trees following Leo Breiman's random forest algorithm. Linear combinations of inputs are derived using `glmnet` in `obliqueRSF-net`, using Newton Raphson scoring for the Cox partial likelihood function in `aorsf-fast` (1 iteration of scoring) and `aorsf-cph` (up to 20 iterations), and chosen randomly from a uniform distribution in `aorsf-extratrees`. Cut-points are selected to maximize a log-rank statistic.

Boosting ensembles			
Trees	`xgboost`	`xgboost-cox` `xgboost-aft`	`xgboost-cox` maximizes the Cox partial likelihood function, whereas `xgboost-aft` maximizes the accelerated failure time likelihood function. Nested cross validation (5 folds) is applied to tune the number of trees. The minimum number of observations in a leaf node was 10, the maximum depth of trees was 6, and $\sqrt{p}$ variables were considered randomly for each split, where p is the number of predictors.
Regression models			
Cox Net	`glmnet`	`glmnet-cox`	The Cox proportional hazards model is fit using an elastic net penalty. Nested cross validation (5 folds) is applied to tune penalty terms.
Neural networks			
Cox Time	`survivalmodels`	`nn-cox`	A neural network based on the proportional hazards model with time-varying effects. Nested cross-validation was applied to select the number of layers (from 1 to 8), the number of nodes in each layer (from $\sqrt{p}/2$ to $\sqrt{p}$), and the number of epochs to complete (up to 500). A drop-out rate of 10% was applied during training.

Table 1: Learning algorithms assessed in numeric studies. `aorsf-fast` is the accelerated oblique random survival forest (see Algorithm 1), and each of the additional learners are compared to `aorsf-fast` in numeric studies.

3.1.2 Evaluation of prediction accuracy

Our primary metric for evaluating the accuracy of predicted risk is the integrated and scaled Brier score (Graf et al., 1999), a proper scoring rule that combines discrimination and calibration in one value and improves interpretability by adjusting for a benchmark model (Kattan and Gerds, 2018). Consider a testing data set:

$$\mathcal{D}_{\text{test}} = \{(T_i, \delta_i, x_i)\}_{i=1}^{N_{\text{test}}}.$$

Let $\widehat{S}(t \mid x_i)$ be the predicted probability of survival up to a given prediction time of $t > 0$. For observation i in $\mathcal{D}_{\text{test}}$, let $\widehat{S}(t \mid \boldsymbol{x}_i)$ be the predicted probability of survival up to a given prediction time of $t > 0$. Define

$$\widehat{\text{BS}}(t) = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \{\widehat{S}(t \mid \boldsymbol{x}_i)^2 \cdot I(T_i \leq t, \delta_i = 1) \cdot \widehat{G}(T_i)^{-1} + [1 - \widehat{S}(t \mid \boldsymbol{x}_i)]^2 \cdot I(T_i > t) \cdot \widehat{G}(t)^{-1}\}$$

where $\widehat{G}(t)$ is the Kaplan-Meier estimate of the censoring distribution. As $\widehat{\text{BS}}(t)$ is time dependent, integration over time provides a summary measure of performance over a range of plausible prediction times. The integrated $\widehat{\text{BS}}(t)$ is defined as

$$\widehat{\mathcal{BS}}(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \widehat{\text{BS}}(t) dt. \tag{2}$$

In our results, t_1 and t_2 are the 25th and 75th percentile of event times, respectively. $\widehat{\mathcal{BS}}(t_1, t_2)$, a sum of squared prediction errors, can be scaled to produce a measure of explained residual variation (*i.e.*, an R^2 statistic) by computing

$$R^2 = 1 - \frac{\widehat{\mathcal{BS}}(t_1, t_2)}{\widehat{\mathcal{BS}}_0(t_1, t_2)} \tag{3}$$

where $\widehat{\mathcal{BS}}_0(t_1, t_2)$ is the integrated Brier score when a Kaplan-Meier estimate for survival based on the training data is used as the survival prediction function $\widehat{S}(t)$. We refer to this R^2 statistic as the index of prediction accuracy (IPA) (Kattan and Gerds, 2018).

Our secondary metric for evaluating predicted risk is the time-dependent concordance (C)-statistic. We compute the first time-dependent C-statistic proposed by Blanche et al. (2013, Equation 3), which is interpreted as the probability that a risk prediction model will assign higher

risk to a case (*i.e.*, an observation with $T \leq t$ and $\delta = 1$) versus a non-case (*i.e.*, an observation with $T > t$). Similar to the IPA, observations with $T \leq t$ and $\delta = 0$ only contribute to inverse probability of censoring weights for the time-dependent C-statistic.

Both the IPA and time-dependent C-statistic generally take values between 0 and 1. To avoid presenting an excessive amount of leading zeroes in our tables, figures, and text, we scale both the IPA and time-dependent C-statistic by 100. For example, we present a value of 25 if the IPA is 0.25, 87 if the time-dependent C-statistic is 0.87, and present 10.2 if the difference between two IPA values is 0.102

3.1.3 Data sets

We used a collection of 16 data sets containing a total of 21 risk prediction tasks (tasks per data set ranged from one to four). Participant-level data from the GUIDE-IT and SPRINT clinical trials and the ARIC, MESA, and JHS community cohort studies was obtained from the National Institute of Health Biologic Specimen and Data Repository Coordinating Center (BioLINCC). Designs and protocols for these studies have been made available (ARIC Investigators, 1989; Bild et al., 2002; Felker et al., 2017; SPRINT Research Group, 2015; Taylor Jr et al., 2005). All other datasets were publicly available and obtained through R packages (see Appendix A.1). Across all prediction tasks, the number of observations ranged from 137 to 17,549 (median: 929), the number of predictors ranged from 7 to 1,692 (median: 12), and the percentage of censored observations ranged from 5.26 to 97.7 (median: 57.3) (Table A.1).

3.1.4 Monte-Carlo cross validation

For each risk prediction task, we completed 25 runs of Monte-Carlo cross validation. In each run, we used a random sample containing 50% of the available data for training and the remaining 50% for testing each of the learners described in Section 3.1.1. Then, for each learner, we computed the IPA and time-dependent C-statistic. If any learner failed to obtain predictions on any particular split of data[5], the results for that split were omitted from downstream analyses for all learners.

[5]For example, when the prediction task was to predict risk of death in the ACTG 320 clinical trial (26 events total), some splits did not leave enough events in the training data to fit complex learners such as neural networks

3.1.5 Statistical analysis

After collecting data from 25 replications of Monte-Carlo cross validation for the 14 learners in all 21 risk prediction tasks, we analyzed the resulting 7,350 observations of IPA and, separately, time-dependent C-statistic, using a Bayesian linear mixed model. Our approach follows the ideas described by Benavoli et al. (2017) and Kuhn and Wickham (2020), who developed guidelines on making statistical comparisons between learners using Bayesian models. Specifically, we fit two models:

$$\text{IPA} = \widehat{\gamma}_0 + \widehat{\gamma} \cdot \text{learner} + (1 \,|\, \text{data/run})$$

and

$$\text{C-stat} = \widehat{\gamma}_0 + \widehat{\gamma} \cdot \text{learner} + (1 \,|\, \text{data/run}).$$

Random intercepts for specific splits of data (*i.e.*, `run` in the model formula) were nested within datasets. The intercept, $\widehat{\gamma}_0$, was the expected value of the outcome using `aorsf-fast`, making the coefficients in $\widehat{\gamma}$ the expected differences between `aorsf-fast` and other learners. Default priors from `rstanarm` were applied for model fitting (Goodrich et al., 2022).

Hypothesis testing For both the IPA and time-dependent C-statistic, we conducted equivalence and inferiority tests based on a 1 point region of practical equivalence. More specifically, we concluded that two learners had practically equivalent IPA or time-dependent C-statistic if there was a 95% or higher posterior probability that the absolute difference in the relevant metric was less than 1. We concluded that one learner was weakly superior when there was $\geq$ 95% posterior probability that the absolute difference in the relevant metric was non-zero, and concluded superiority when when there was $\geq$ 95% posterior probability that the absolute difference in the relevant metric was 1 or more.

3.2 Benchmark of variable importance

The aim of this experiment is to evaluate negation VI and similar VI methods based on how well they can discriminate between relevant and irrelevant variables, where relevance is defined by having a relationship with the simulated outcome. We consider methods that are intrinsic to the oblique RF (*e.g.*, ANOVA VI), those that are intrinsic to the RF (*e.g.*, permutation VI), and those that are

model-agnostic (*e.g.*, SHAP VI). VI methods with unavailable or still developing software were not included.[6]

3.2.1 Variable importance techniques

We compute permutation VI for axis based RSFs using the `randomForestSRC` package. We compute ANOVA VI, negation VI, and permutation VI for oblique RSFs using the `aorsf` package. For ANOVA VI, we applied a p-value threshold of 0.01, following the threshold recommended by Menze et al. (2011). We compute SHAP VI for boosted tree models using the `xgboost` package (Chen et al., 2022), which incorporates the tree SHAP approach proposed by Lundberg et al. (2018).

3.2.2 Variable types

We considered five classes of predictor variables, with each class characterized by its variables' relationship to a right-censored outcome on the log-hazard scale. Specifically,

- *irrelevant* variables had no relationship with the outcome.
- *main effect* variables had a linear relationship to the outcome on the log-hazard scale.
- *non-linear effect* variables had a non-linear relationship to the outcome. A normally distributed variable x was generated with a linear relationship to the outcome on the log-hazard scale, then $\tilde{x} = \sin(a \cdot \pi \cdot x)$ was retained for modeling. The constant a varied uniformly from 0.125 to 0.25.
- *combination effect* variables were formed by linear combinations of three other variables. While their combination was linearly related to the outcome on the log-hazard scale, each of the three variables contributing to the combination had no relation to the outcome.
- *interaction effect* variables were related to the outcome by multiplicative interaction with one other variable, which could have been a main effect, non-linear effect, or combination effect variable.

[6]Although the `party` package implements the approach to VI developed by Strobl et al. (2007), the developers of the `party` package note that the implementation of this approach for survival outcomes is "extremely slow and experimental" as of version 1.3.10. Therefore, it is not incorporated in the current simulation study.

3.2.3 Simulated data

We initiated each set of simulated data with a random draw of size n from a p-dimensional multivariate normal distribution, yielding n observations of p predictors. Each of p predictor variables had a mean of zero, standard deviation of 1, and correlation with other predictor variables drawn at random between a lower and upper boundary. A time-to-event outcome with roughly 45% of observations censored was generated using the `simsurv` package (Brilleman, 2018; Brilleman et al., 2020). The full predictor matrix (*i.e.*, including interactions, non-linear mappings, and combinations) was used to generate the outcome. Interactions, non-linear mappings, and combinations were dropped from the predictor matrix after the outcome was generated so that VI techniques could be evaluated based on their ability to detect these effects.

3.2.4 Parameter specifications

Parameters that varied in the current simulation study included the number of observations (500, 1000, and 2500) and the absolute value of the maximum correlation between predictors (0.3, 0.15, and 0). Parameters that remain fixed throughout the study included the number of predictors in each class (15) and the effect size of each predictor (one standard deviation increase associated with a 64% increase in relative risk). Using this design for simulated data, the Heller explained relative risk (95% confidence interval) of our covariates was 88.5 (88.2, 88.7) (Heller, 2012) with 2,500 observations.

3.2.5 Evaluation of variable importance

We compared VI techniques based on their discrimination (*i.e.*, C-statistic) between relevant and irrelevant variables. Specifically, we generated a binary outcome for each predictor variable based on its relevance (*i.e.*, the binary outcome is 1 if the variable is relevant, 0 otherwise). Treating VI as if it were a 'prediction' for these binary outcomes yields a C-statistic which may be interpreted as the probability that the VI technique will rank a relevant variable higher than an irrelevant variable (Harrell et al., 1982).

3.3 Benchmark of computational efficiency

The aim of this numeric experiment is to evaluate the computational efficiency of the accelerated oblique RSF compared to its predecessor (the oblique RSF from the `obliqueRSF` R package) and to several other machine learning algorithms.

3.3.1 Evaluation of computational efficiency

For each learner discussed in Section 3.1.1 and for each of the 21 risk prediction tasks analyzed in Section 3.1, we tracked the amount of time required to fit a prediction model (including time used to tune parameters) and compute predicted risk.

We performed additional benchmarks on the time required to fit 500 trees using `aorsf`, `randomForestSRC`, and `ranger`. The learners that represented these R packages were `aorsf-fast`, `rsf-standard`, and `rsf-extratrees`, respectively. To allow for controlled comparisons of computational efficiency with varying dimensions of training data, we used the same process to simulate data as described in Section 3.2.3, varying the number of observations from 100 to 10000 and the number of predictors from 10 to 1000. The minimum node size of trees in this experiment was dynamically set as the nearest integer to the number of observations in the training set divided by 10.

3.4 Computational details

All analyses were conducted using R version 4.1.3 with version 0.0.4 of the `aorsf` (Jaeger et al., 2022) package. Analyses were coordinated with assistance from the `targets` package (Landau, 2021). To standardize comparisons of computational efficiency, all learners and VI techniques used up to 4 processing units.

4 Results

In Section 4.1, Section 4.2, and Section 4.3, we present results from the benchmark or prediction accuracy, the simulation study of VI, and the benchmark of computational efficiency, respectively.

4.1 Prediction accuracy

A full summary of all results presented in this Section is provided in Table A.2. In total, 521 out of 525 Monte-Carlo cross validation runs were completed. On run 13, 18, 24 and 25 for the ACTG 320 data, the `nn-cox` learner encountered an error during its fitting procedure.

Index of prediction accuracy Compared to learners that were not oblique RSFs, `aorsf-fast` had the highest IPA in 7 out of 21 risk prediction tasks, with an overall mean IPA of 14.1 (Figure 4). Compared to the learner with the second highest mean IPA (`cif-rotate`), `aorsf-fast`'s mean was 0.851 points higher, a relative increase of 6.43%. The posterior probability of `aorsf-fast` and `aorsf-cph` having practically equivalent expected IPA was 0.89, and the posterior probability of `aorsf-fast` having a superior IPA to other learners ranged from 0.37 (versus `cif-rotate`) to >0.999 (versus several other learners; see Figure 5)

Time-dependent concordance statistic Compared to learners that were not oblique RSFs, `aorsf-fast` had the highest time-dependent C-statistic in 6 out of 21 risk prediction tasks, with an overall mean of 75.8 (Figure 6). Compared to the learner with the second highest mean C-statistic (`cif-standard`), `aorsf-fast`'s mean was 0.903 points higher, a relative increase of 1.20%. The posterior probability of `aorsf-fast` and `aorsf-cph` having practically equivalent expected time-dependent C-statistics was 0.96, and the posterior probability of `aorsf-fast` having a superior time-dependent C-statistic versus other learners ranged from 0.44 (versus `cif-standard`) to >0.999 (versus several other learners; see Figure 7)

4.2 Variable importance

The three techniques that used 'aorsf' to estimate VI were ranked first (`aorsf-negate`; $C = 75.9$), second (`aorsf-anova`; $C = 73.9$), and third (`aorsf-permute`; $C = 73.2$) in overall mean C-statistic across all of the simulation scenarios, with `aorsf-negate` obtaining the highest C-statistic in 26 out of 36 VI tasks (Figure 8). Among the four relevant variable classes, `aorsf-negate` had the highest mean C-statistic for main effects, combination effects, and non-linear effects, with the greatest advantage of using `aorsf-negate` occurring among non-linear and combination variables. Full results from the experiment are provided in Table A.3. Computationally, ANOVA VI was faster

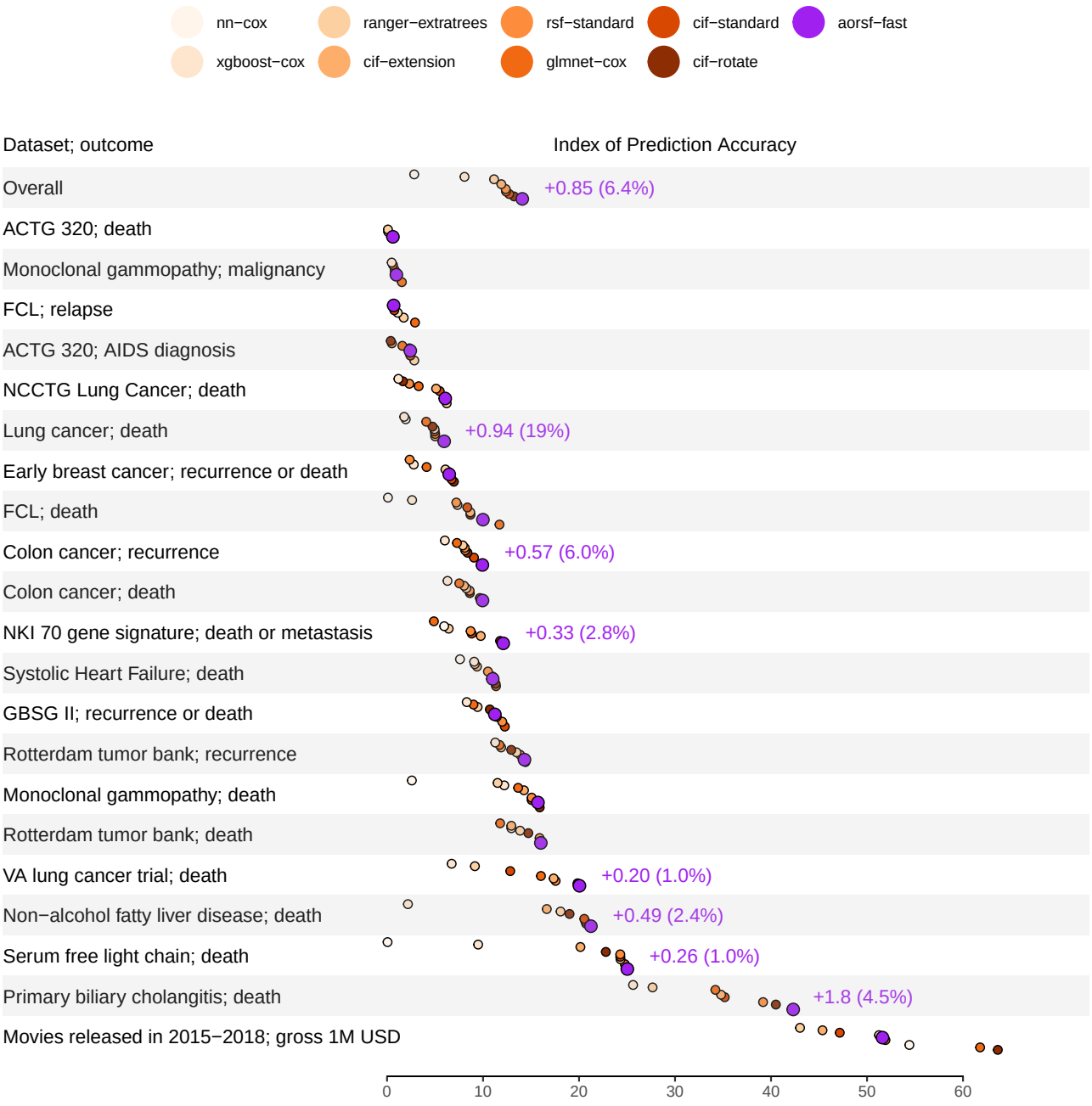

Figure 4: Index of prediction accuracy in multiple risk prediction tasks. Text appears in tasks where the accelerated oblique random survival forest obtained the highest score, showing absolute and relative improvement over the second best learner. Since this figure is intended to compare `aorsf-fast` to learners that are not oblique random survival forests, `aorsf-cph`, `aorsf-random`, and `obliqueRSF-net` are not included.

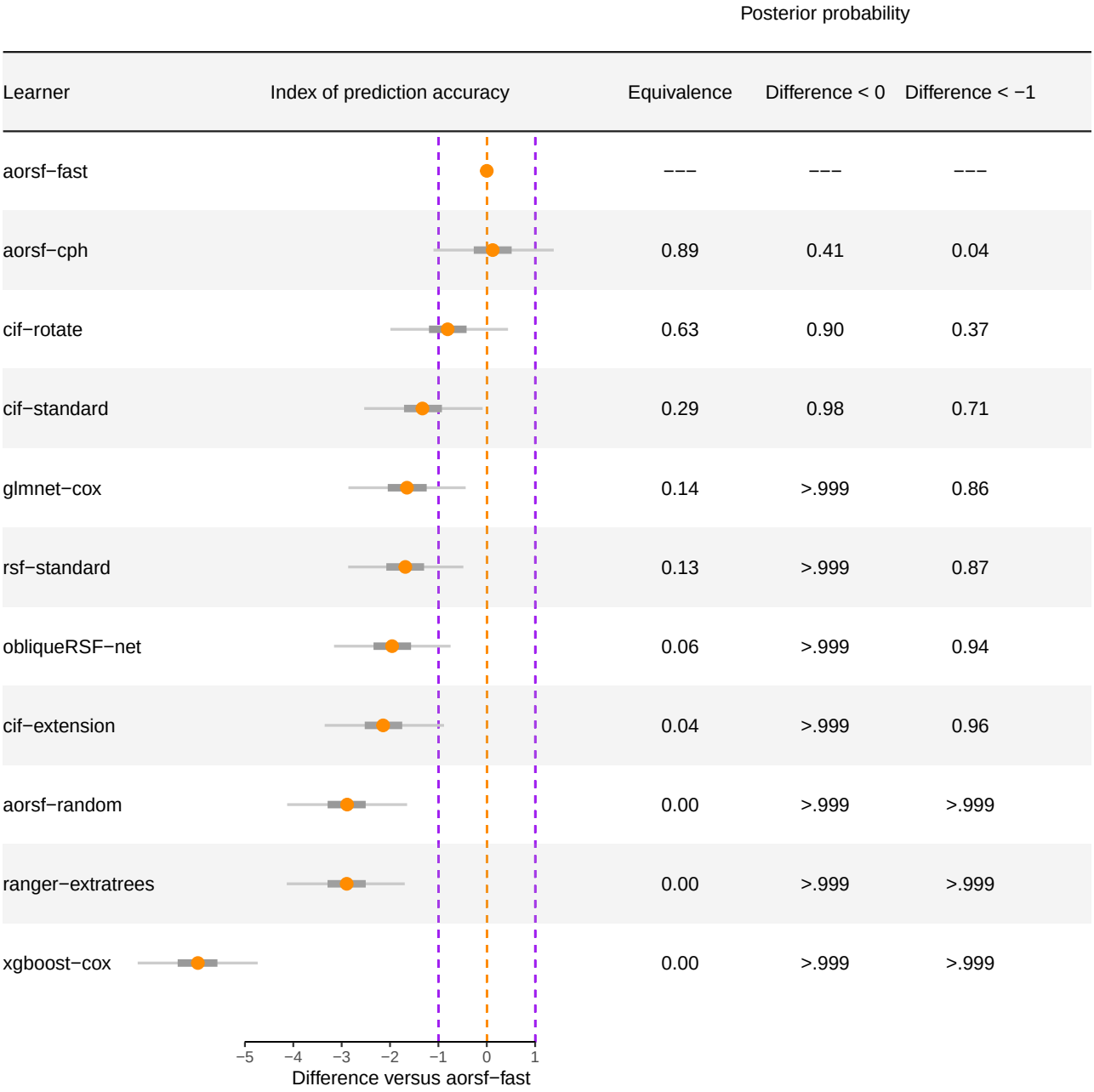

Figure 5: Expected differences in index of prediction accuracy between the accelerated oblique random survival forest and other learning algorithms. A region of practical equivalence is shown by purple dotted lines, and a boundary of non-zero difference is shown by an orange dotted line at the origin.

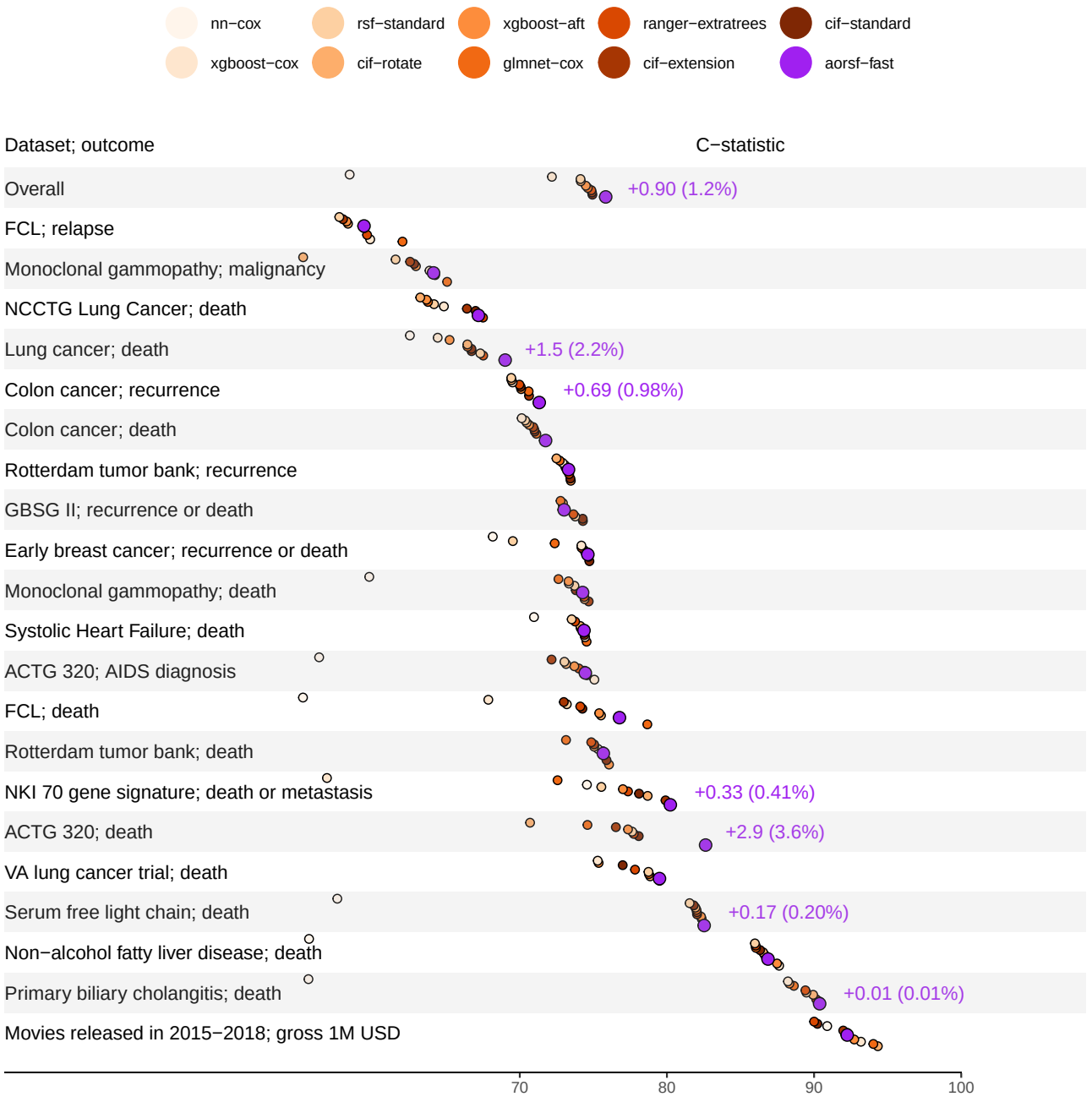

Figure 6: Time-dependent concordance statistic for the accelerated oblique random survival forest and other learning algorithms across multiple risk prediction tasks. Text appears in tasks where the accelerated oblique random survival forest obtained the highest concordance, showing the absolute and percent improvement over the second best learner. Since this figure is intended to compare `aorsf-fast` to learners that are not oblique random survival forests, `aorsf-cph`, `aorsf-random`, and `obliqueRSF-net` are not included.

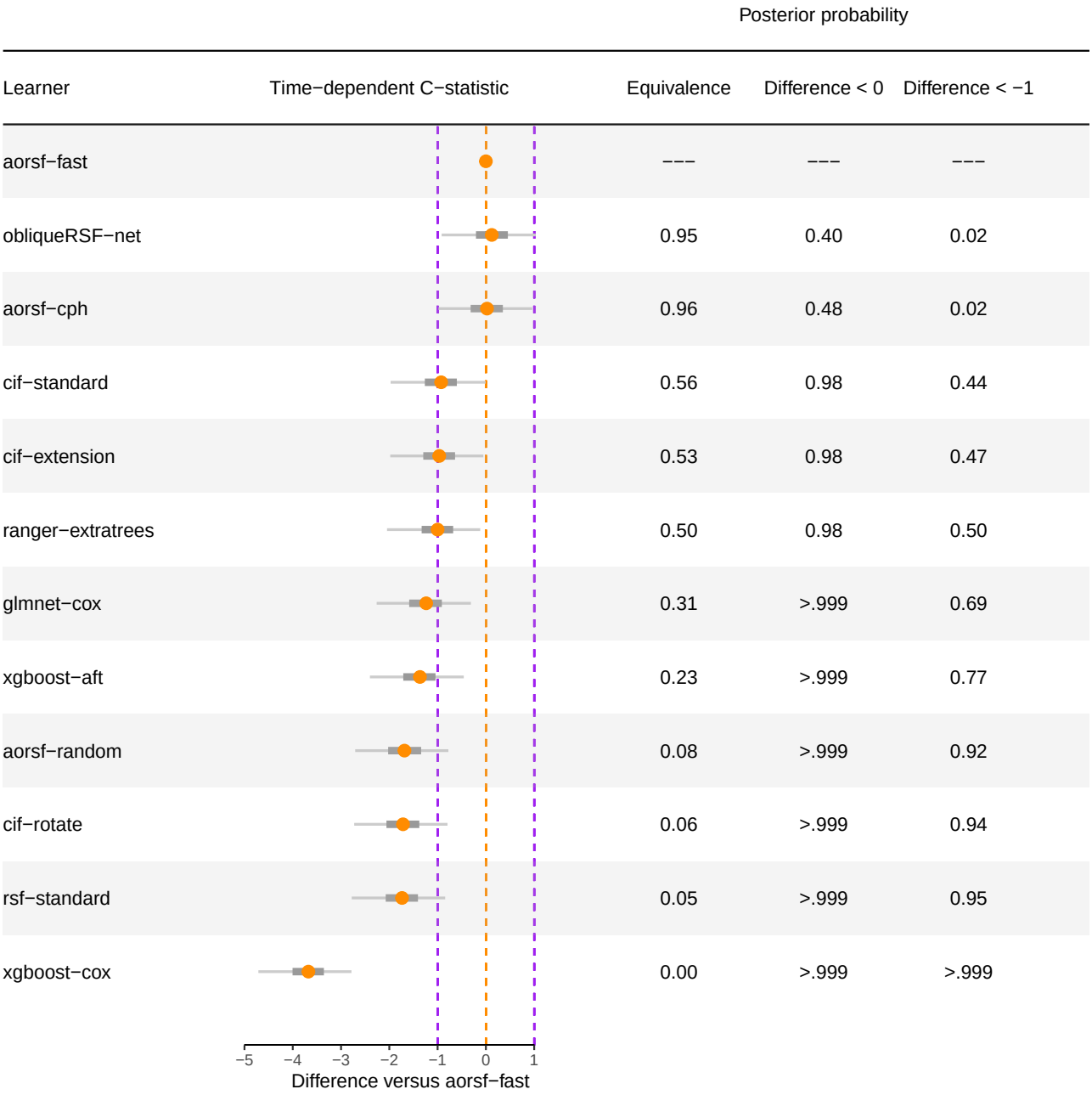

Figure 7: Expected differences in time-dependent concordance statistic between the accelerated oblique random survival forest and other learning algorithms. A region of practical equivalence is shown by purple dotted lines, and a boundary of non-zero difference is shown by an orange dotted line at the origin.

than negation and permutation VI, with a median time of 2.88 seconds versus 20.4 and 21.8 seconds, respectively.

4.3 Computational efficiency

In the analysis of 21 risk prediction tasks, `aorsf-fast` was the second fastest learner overall, with a median time to develop a risk prediction model and compute predictions about 76 milliseconds longer than `glmnet-cox` (Figure 9). Comparing median computing times, `aorsf-fast` was 917.1 times faster than its predecessor, `obliqueRSF-net`. In addition, `aorsf-fast` was 16.0, 0.820, and 4.70 faster than axis based forests grown using the `party`, `ranger`, and `randomForestSRC` packages, respectively.

In the analysis of time to fit 500 trees using simulated data, the `ranger` package exhibited the fastest computation times overall (Figure 10). `aorsf` was the second fastest when the number of predictors was 10 or 100, and `randomForestSRC` had similar computation time versus `aorsf` when 1000 predictors were present.

5 Discussion

In this paper, we have developed two contributions to the oblique RSF: (1) the accelerated oblique RSF (*i.e.*, `aorsf-fast`) and (2) negation VI. Our technique to accelerate the oblique RSF reduces the number of operations required to find linear combinations of inputs using a single iteration of Newton Raphson scoring, while our VI technique directly engages with coefficients in linear combinations of inputs to measure importance of individual variables. In numeric experiments, we found that `aorsf-fast` is approximately 917.1 times faster than its predecessor, `obliqueRSF-net`, with a practically equivalent C-statistic. We also found that negation VI, a technique to estimate VI using the oblique RSF, detected non-linear, combination, and main effects more effectively than three standard methods to estimate VI: permutation, ANOVA, and SHAP VI. Overall, we found that estimating VI using negation instead of ANOVA increased the C-statistic for ranking a relevant variable higher than an irrelevant variable by 2.05, a relative increase of 2.78%.

To understand potential differences in computational efficiency, we reviewed code in the `aorsf`, `randomForestSRC`, and `ranger` packages. We found differences in how survival outcome data are

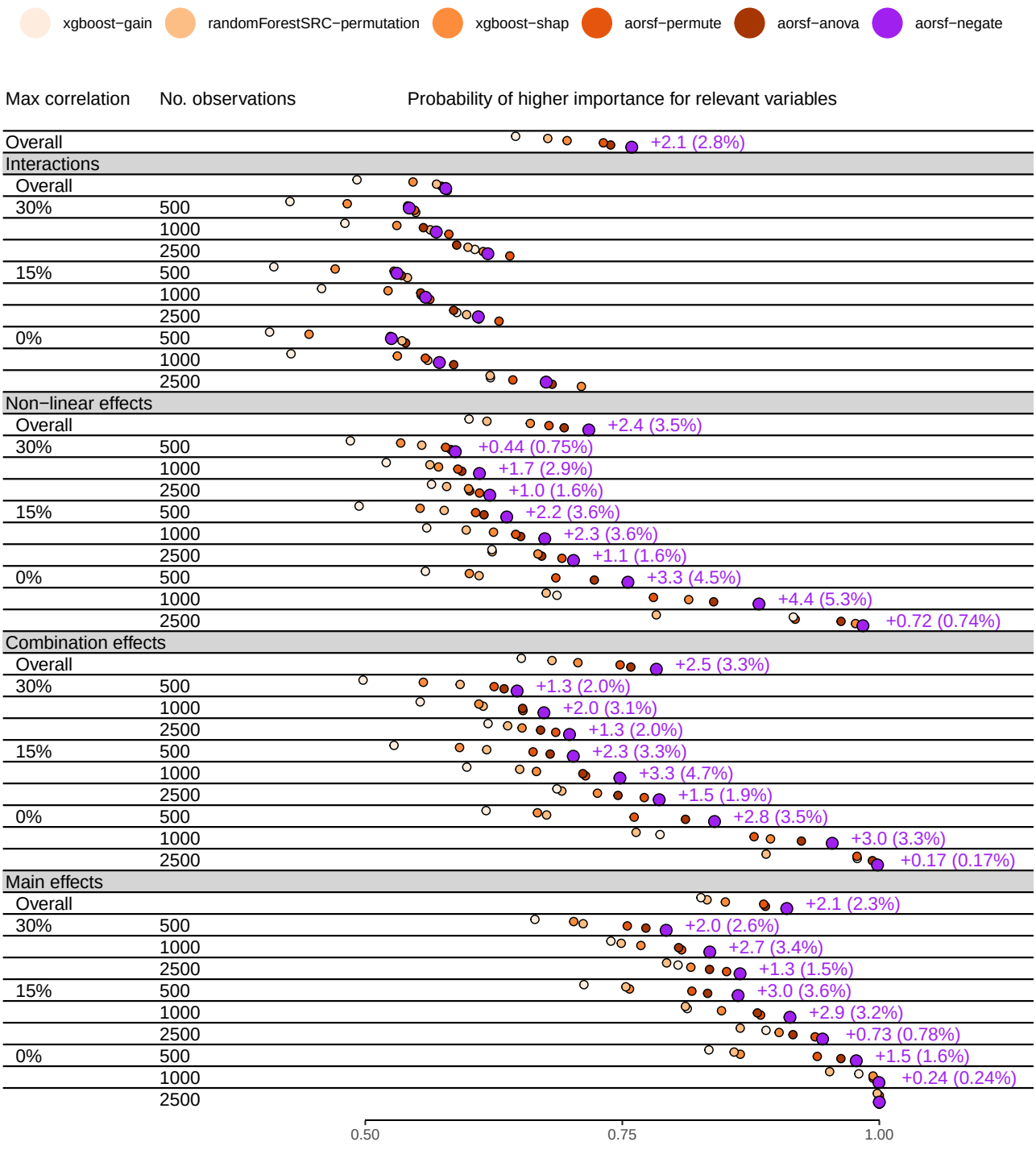

Figure 8: Concordance statistic for assigning higher importance to relevant versus irrelevant variables. Text appears in rows where negation importance obtained the highest concordance, showing absolute and percent improvement over the second best technique.

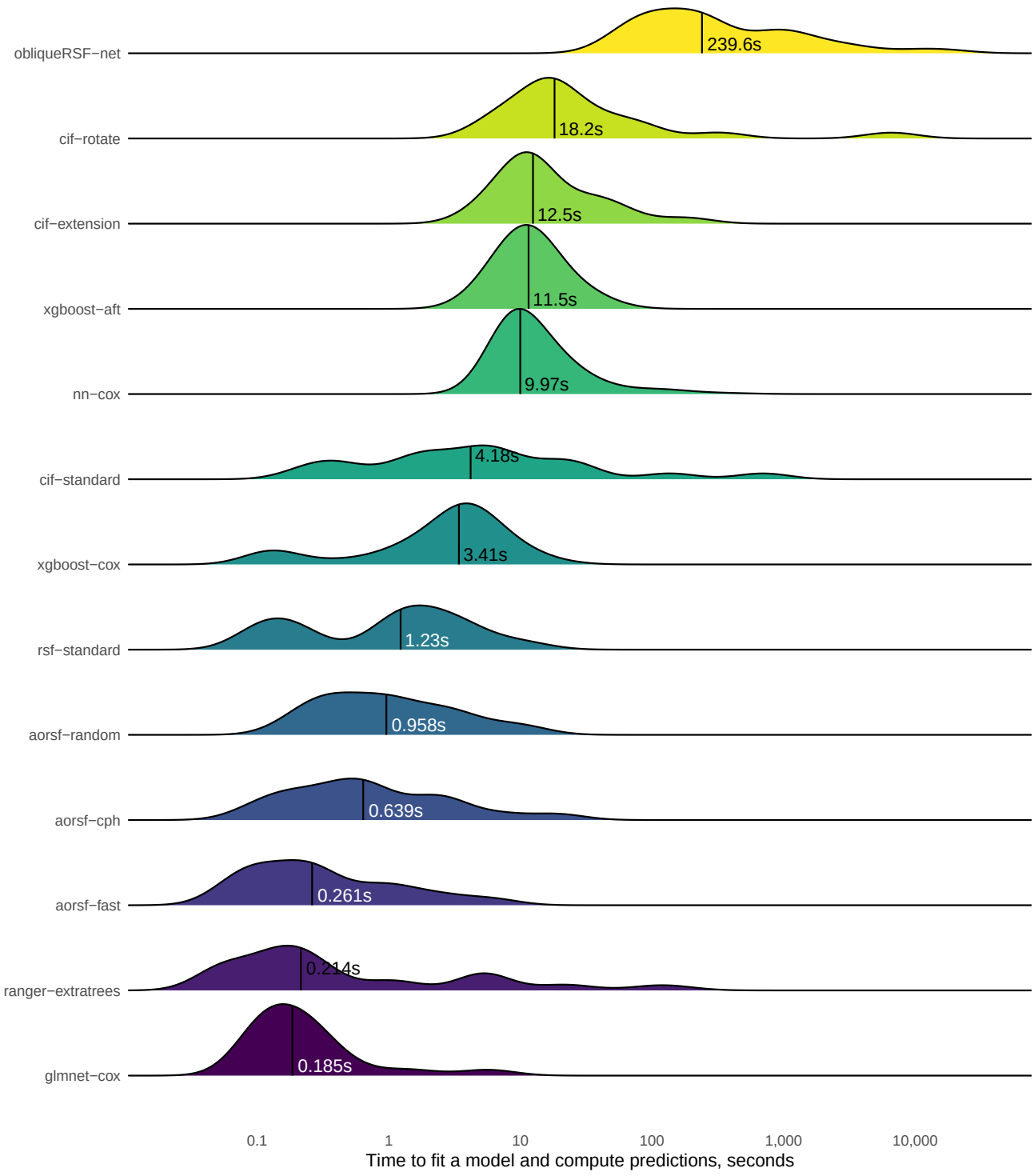

Figure 9: Distribution of time taken to fit a prediction model and compute predicted risk. The median time, in seconds, is printed and annotated for each learner by a vertical line.

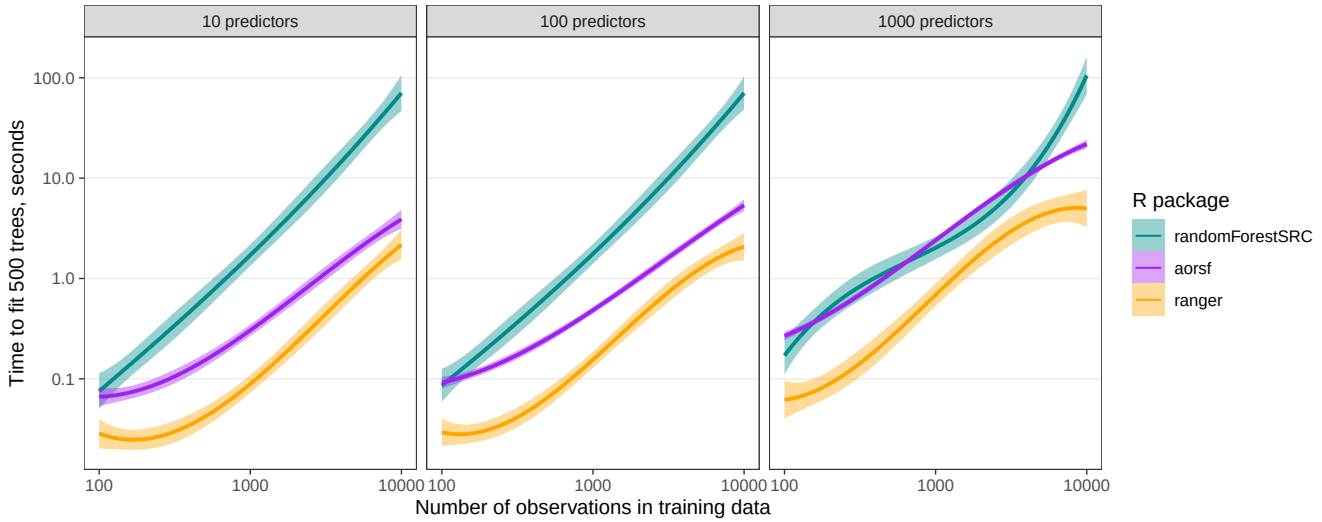

Figure 10: The expected time, in seconds, to fit an ensemble of 500 axis-based survival trees using the ranger or randomForestSRC package versus 500 oblique survival trees using the aorsf package. The ranger package is the most efficient overall, and aorsf appears to be relatively efficient in larger samples, particularly when 10 or 100 predictors are present in the training data. All three packages appear to scale linearly in computation time with the number of observations in the training data.

saved in leaf nodes. For each leaf node, `aorsf` stores data with one row per unique event time using training data that are stored in the leaf, whereas `randomForestSRC` and `ranger` store survival outcomes at a fixed grid of event times in each leaf. By default, `ranger` creates a grid that includes all event times in the training data. The grid strategy can cause higher computing time and memory usage when the grid of event times is large and a large number of leaf nodes are included in each tree, which can occur when minimum node size is small relative to the size of the training data. We kept minimum node size fixed in our benchmark of computational efficiency using real data, and dymanically increased minimum node size based on the size of the training set when we benchmarked computational efficiency using simulated data. Because of this decision, the `randomForestSRC` and `ranger` packages ran slower than `aorsf` in our benchmark of real data but not in the benchmark of simulated data.

5.1 Implications of our results

Accurate risk prediction models have the potential to improve healthcare by directing timely interventions to patients who are most likely to benefit. However, prediction models that cannot scale adequately to large databases or cannot be interpreted and explained will struggle to gain acceptance in clinical practice (Moss et al., 2022). The current study advances the oblique RSF, an accurate risk prediction model, towards being accurate, scalable, and interpretable. The improved computational efficiency of the accelerated oblique RSF increases the feasibility of applying oblique RSFs in a wide range of prediction tasks. Faster model evaluation and re-fitting also improve diagnosis and resolution of model-based issues (*e.g.*, model calibration deteriorates over time). The introduction of negation VI also advances interpretability. VI is intrinsically linked to model fairness, as it can be used to identify when protected characteristics such as race, religion, and sexuality are inadvertently used (either directly or through correlates of these characteristics) by a prediction model. Since negation VI engages with the coefficients used in linear combinations of variables, a major component of oblique RSFs, it may be more capable of diagnosing unfairness in oblique RSFs compared to permutation importance and model-agnostic VI techniques.

5.2 Limitations and next steps

While the current study advances the oblique RSF towards being scalable and interpretable, there remain several limitations that can be targeted in future studies. The accelerated oblique RSF does not account for competing risks, and biased estimation of incidence may occur when competing risks are ignored. Thus, allowing the oblique RSF to account for competing risks is a high priority for future studies. In addition, the current study only considered data without missing values, only evaluated oblique RSFs that applied the log-rank statistic for node splitting, and only considered negation VI estimates based on Harrell's C-statistic. Few studies have developed strategies to deal with missing data while growing oblique survival trees. Prior studies have found that log-rank tests can be mis-informative when survival curves cross (Li et al., 2015), and that Harrell's C-statistic is dependent on the censoring distribution of the outcome (Uno et al., 2011). Thus, a second item is to expand the range of options available to users of the `aorsf` package, enabling them to apply strategies for imputation of missing values and use a broad range of statistical criteria while growing

oblique survival trees. Last, Cui et al. (2017) found that estimating an inverse-probability weighted hazard function at each non-leaf node of a survival tree allows the RSF to converge asymptotically to the true survival function when some variables contribute both to the risk of the event and the risk of censoring, a scenario that is very likely in the analysis of medical data. The accelerated oblique RSF could incorporate this splitting technique by using Newton Raphson scoring to fit a model for the censoring distribution after which a weighted model could be fit to the failure distribution. This final item has the highest priority, as Cui et al. (2017) showed it is a requisite condition for consistency of axis-based survival trees in fairly general settings.

5.3 Conclusion

Oblique RSFs have exceptional prediction accuracy and this study has shown how they can be fit with computational efficiency that rivals their axis-based counterparts. We have also introduced a general and flexible method to estimate VI with oblique RFs, and demonstrated its effectiveness for numeric, correlated predictors.

Appendix

Data sources

1. The "VA lung cancer trial" data (Kalbfleisch and Prentice, 2011) were obtained from the `randomForestSRC` R package (Ishwaran and Kogalur, 2019).

2. The "Colon cancer" data (Moertel et al., 1995) were obtained from the `survival` R package (Therneau, 2022b).

3. The "Primary biliary cholangitis" data (Therneau and Grambsch, 2000) were obtained from the `aorsf` R package (Jaeger, 2022).

4. The "Movies released in 2015-2018" data were obtained from the `censored` R package (Hvitfeldt and Frick, Hvitfeldt and Frick).

5. The "GBSG II" data (Schumacher, 1994) were obtained from the `TH.data` R package (Hothorn, 2022).

6. The "Systolic Heart Failure" data (Hsich et al., 2011) were obtained from the `randomForestSRC` R package (Ishwaran and Kogalur, 2019).

7. The "Serum free light chain" data (Dispenzieri et al., 2012; Kyle et al., 2006) were obtained from the `survival` R package (Therneau, 2022b).

8. The "Non-alcohol fatty liver disease" data (Allen et al., 2018) were obtained from the `survival` R package (Therneau, 2022b).

9. The "Rotterdam tumor bank" data (Royston and Altman, 2013) were obtained from the `survival` R package (Therneau, 2022b).

10. The "ACTG 320" data (Hosmer and Lemeshow, 2002) were obtained from the `mlr3proba` R package (Sonabend et al., 2021).

11. The "Early breast cancer" data (Desmedt et al., 2011; Hatzis et al., 2011; Ternès et al., 2017) were obtained from the `biospear` R package (Ternes et al., 2018).

12. The "NKI 70 gene signature" data (Van De Vijver et al., 2002) were obtained from the `OpenML` R package (Casalicchio et al., 2017).

13. The "Lung cancer" data (Director's Challenge Consortium for the Molecular Classification of Lung Adenocarcinoma, 2008) were obtained from the `OpenML` R package (Casalicchio et al., 2017).

14. The "NCCTG Lung Cancer" data (Loprinzi et al., 1994) were obtained from the `survival` R package (Therneau, 2022b).

15. The "FCL" data (Pintilie, 2006) were obtained from the `randomForestSRC` R package (Ishwaran and Kogalur, 2019).

16. The "Monoclonal gammopathy" data (Kyle et al., 2002) were obtained from the `survival` R package (Therneau, 2022b).

A.1: Data sets used for numeric experiments

Label	N observations	N predictors	Outcome	N Events	% Censored
VA lung cancer trial	137	8	Death	128	6.57
Colon cancer	929	12	Recurrence	468	49.6
			Death	452	51.3
Primary biliary cholangitis	276	19	Death	111	59.8
Movies released in 2015-2018	551	46	Gross 1M USD	522	5.26
GBSG II	686	10	Recurrence Or Death	299	56.4
Systolic Heart Failure	2,231	41	Death	726	67.5
Serum free light chain	7,874	10	Death	2,169	72.5
Non-alcohol fatty liver disease	17,549	24	Death	1,364	92.2
Rotterdam tumor bank	2,982	11	Recurrence	1,518	49.1
			Death	1,272	57.3
ACTG 320	1,151	12	AIDS Diagnosis	96	91.7
			Death	26	97.7
Early breast cancer	614	1,692	Recurrence Or Death	134	78.2
NKI 70 gene signature	144	77	Death Or Metastasis	48	66.7
Lung cancer	442	24	Death	236	46.6

NCCTG Lung Cancer	228	9	Death	165	27.6
FCL	541	7	Death	76	86.0
			Relapse	272	49.7
Monoclonal gammopathy	1,384	8	Death	963	30.4
			Malignancy	115	91.7

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks.

	Performance metric (SD)		Computation time, seconds	
	Scaled Brier	C-Statistic	Model fitting	Risk prediction
Overall				
aorsf-cph	0.142 (0.129)	0.759 (0.083)	0.598	0.042
aorsf-fast	0.141 (0.129)	0.758 (0.084)	0.200	0.043
cif-rotate	0.132 (0.153)	0.741 (0.096)	14.486	3.435
cif-standard	0.127 (0.116)	0.749 (0.082)	0.939	3.370
glmnet-cox	0.124 (0.144)	0.746 (0.087)	0.183	0.002
rsf-standard	0.124 (0.135)	0.741 (0.088)	1.103	0.105
obliqueRSF-net	0.121 (0.094)	0.760 (0.081)	211.788	20.351
cif-extension	0.119 (0.111)	0.749 (0.086)	8.578	3.809
aorsf-random	0.111 (0.092)	0.742 (0.076)	0.915	0.040
ranger-extratrees	0.111 (0.101)	0.749 (0.079)	0.057	0.143
xgboost-cox	0.081 (0.121)	0.722 (0.108)	3.406	0.003
nn-cox	0.028 (0.124)	0.584 (0.126)	9.229	0.680
xgboost-aft	—	0.745 (0.088)	11.525	0.006
ACTG 320; AIDS diagnosis, n = 1151, p = 12				
obliqueRSF-net	0.029 (0.015)	0.753 (0.037)	115.320	18.196
ranger-extratrees	0.028 (0.017)	0.740 (0.036)	0.086	0.133
aorsf-random	0.027 (0.020)	0.756 (0.036)	0.465	0.035
cif-standard	0.024 (0.031)	0.744 (0.040)	1.657	4.377
aorsf-cph	0.024 (0.029)	0.750 (0.042)	0.436	0.036
aorsf-fast	0.024 (0.028)	0.745 (0.045)	0.141	0.036
cif-extension	0.023 (0.015)	0.722 (0.038)	9.010	4.189
glmnet-cox	0.016 (0.030)	0.746 (0.037)	0.197	0.002

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
rsf-standard	0.005 (0.041)	0.730 (0.042)	0.179	0.061
cif-rotate	0.004 (0.040)	0.731 (0.038)	14.549	3.604
nn-cox	0.000 (0.011)	0.564 (0.101)	7.755	0.811
xgboost-cox	-0.001 (0.052)	0.751 (0.033)	3.729	0.003
xgboost-aft	—	0.737 (0.035)	11.383	0.006
ACTG 320; death, n = 1151, p = 12				
aorsf-random	0.008 (0.012)	0.798 (0.073)	0.285	0.024
obliqueRSF-net	0.006 (0.009)	0.821 (0.049)	49.070	11.201
aorsf-fast	0.006 (0.019)	0.826 (0.057)	0.088	0.020
aorsf-cph	0.006 (0.018)	0.818 (0.062)	0.357	0.020
cif-extension	0.001 (0.020)	0.765 (0.066)	8.283	3.478
ranger-extratrees	0.001 (0.019)	0.777 (0.069)	0.041	0.122
xgboost-cox	-0.004 (0.004)	0.500 (0.000)	0.118	0.002
nn-cox	-0.004 (0.004)	0.547 (0.128)	7.487	0.717
cif-standard	-0.005 (0.025)	0.781 (0.062)	1.695	4.223
rsf-standard	-0.031 (0.051)	0.776 (0.073)	0.098	0.037
cif-rotate	-0.037 (0.049)	0.707 (0.090)	13.163	3.201
glmnet-cox	-0.065 (0.095)	0.746 (0.098)	0.286	0.002
xgboost-aft	—	0.774 (0.070)	10.124	0.005
Colon cancer; death, n = 929, p = 12				
aorsf-random	0.103 (0.011)	0.724 (0.012)	0.974	0.048
aorsf-cph	0.100 (0.015)	0.717 (0.011)	0.631	0.050
aorsf-fast	0.099 (0.014)	0.718 (0.012)	0.235	0.052
cif-standard	0.097 (0.013)	0.710 (0.012)	0.698	3.233

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
obliqueRSF-net	0.087 (0.006)	0.717 (0.011)	227.346	90.031
cif-rotate	0.086 (0.017)	0.705 (0.014)	12.581	3.222
rsf-standard	0.086 (0.019)	0.704 (0.011)	1.899	0.150
ranger-extratrees	0.083 (0.007)	0.710 (0.012)	0.079	0.231
cif-extension	0.080 (0.006)	0.709 (0.011)	7.680	3.708
glmnet-cox	0.075 (0.016)	0.711 (0.019)	0.133	0.002
xgboost-cox	0.063 (0.013)	0.701 (0.013)	3.694	0.003
nn-cox	-0.003 (0.003)	0.510 (0.045)	9.217	1.188
xgboost-aft	—	0.706 (0.013)	12.025	0.006
Colon cancer; recurrence, n = 929, p = 12				
aorsf-fast	0.099 (0.017)	0.713 (0.016)	0.235	0.051
aorsf-cph	0.099 (0.016)	0.712 (0.015)	0.641	0.050
aorsf-random	0.094 (0.014)	0.706 (0.015)	0.989	0.047
cif-standard	0.091 (0.016)	0.701 (0.017)	0.685	3.216
obliqueRSF-net	0.086 (0.008)	0.712 (0.015)	220.136	52.240
cif-rotate	0.084 (0.020)	0.694 (0.017)	12.394	3.355
cif-extension	0.081 (0.009)	0.706 (0.017)	7.829	3.620
rsf-standard	0.081 (0.020)	0.694 (0.015)	1.839	0.152
ranger-extratrees	0.079 (0.011)	0.700 (0.016)	0.081	0.273
glmnet-cox	0.073 (0.018)	0.706 (0.024)	0.136	0.002
xgboost-cox	0.060 (0.010)	0.695 (0.018)	3.234	0.003
nn-cox	-0.020 (0.074)	0.533 (0.044)	9.225	1.019
xgboost-aft	—	0.701 (0.019)	12.802	0.006
Early breast cancer; recurrence or death, n = 614, p = 1692				

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
obliqueRSF-net	0.072 (0.022)	0.751 (0.027)	1772.643	38.287
cif-rotate	0.070 (0.018)	0.747 (0.027)	6243.357	338.140
cif-standard	0.067 (0.019)	0.747 (0.030)	8.875	4.293
aorsf-cph	0.067 (0.029)	0.747 (0.026)	1.614	0.300
aorsf-fast	0.065 (0.028)	0.746 (0.026)	1.325	0.297
cif-extension	0.064 (0.016)	0.746 (0.028)	42.920	6.083
ranger-extratrees	0.061 (0.022)	0.742 (0.031)	0.219	0.311
glmnet-cox	0.041 (0.032)	0.724 (0.036)	5.782	0.005
xgboost-cox	0.028 (0.032)	0.742 (0.032)	2.472	0.006
aorsf-random	0.025 (0.016)	0.691 (0.042)	1.888	0.271
rsf-standard	0.024 (0.037)	0.695 (0.033)	0.883	0.169
nn-cox	-0.010 (0.071)	0.682 (0.067)	14.875	1.621
xgboost-aft	—	0.744 (0.027)	10.373	0.009
FCL; death, n = 541, p = 7				
glmnet-cox	0.117 (0.028)	0.787 (0.037)	0.105	0.002
aorsf-cph	0.100 (0.039)	0.769 (0.033)	0.165	0.018
aorsf-fast	0.100 (0.037)	0.768 (0.033)	0.079	0.018
obliqueRSF-net	0.091 (0.023)	0.769 (0.032)	78.242	6.014
cif-rotate	0.087 (0.048)	0.755 (0.027)	5.839	1.758
cif-extension	0.087 (0.036)	0.730 (0.034)	5.195	2.616
aorsf-random	0.085 (0.029)	0.754 (0.034)	0.258	0.019
cif-standard	0.084 (0.038)	0.743 (0.036)	0.281	1.194
ranger-extratrees	0.073 (0.016)	0.741 (0.037)	0.031	0.081
rsf-standard	0.072 (0.048)	0.732 (0.034)	0.113	0.039
xgboost-cox	0.026 (0.053)	0.679 (0.120)	0.330	0.002

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
nn-cox	0.001 (0.028)	0.553 (0.117)	7.201	0.403
xgboost-aft	—	0.754 (0.038)	7.320	0.005
FCL; relapse, n = 541, p = 7				
glmnet-cox	0.029 (0.017)	0.620 (0.024)	0.107	0.002
obliqueRSF-net	0.018 (0.014)	0.598 (0.024)	165.938	8.270
ranger-extratrees	0.017 (0.016)	0.596 (0.025)	0.031	0.080
aorsf-random	0.012 (0.017)	0.595 (0.023)	0.401	0.021
xgboost-cox	0.011 (0.016)	0.598 (0.032)	1.548	0.002
cif-standard	0.008 (0.021)	0.594 (0.023)	0.277	1.221
aorsf-cph	0.007 (0.021)	0.595 (0.026)	0.260	0.023
aorsf-fast	0.007 (0.019)	0.594 (0.025)	0.116	0.022
cif-extension	-0.005 (0.023)	0.580 (0.028)	5.912	2.183
nn-cox	-0.006 (0.014)	0.521 (0.059)	8.447	0.457
cif-rotate	-0.012 (0.025)	0.583 (0.030)	6.486	2.537
rsf-standard	-0.026 (0.032)	0.577 (0.024)	0.891	0.083
xgboost-aft	—	0.582 (0.034)	6.799	0.005
GBSG II; recurrence or death, n = 686, p = 10				
cif-standard	0.123 (0.020)	0.743 (0.020)	0.478	2.173
obliqueRSF-net	0.121 (0.014)	0.747 (0.018)	234.738	19.092
rsf-standard	0.120 (0.023)	0.738 (0.019)	1.362	0.114
aorsf-cph	0.117 (0.022)	0.733 (0.017)	0.404	0.038
cif-extension	0.114 (0.017)	0.743 (0.019)	7.544	3.429
aorsf-fast	0.112 (0.024)	0.730 (0.018)	0.180	0.040
aorsf-random	0.111 (0.017)	0.727 (0.018)	0.728	0.036

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
cif-rotate	0.107 (0.023)	0.729 (0.017)	10.139	2.954
ranger-extratrees	0.094 (0.018)	0.736 (0.025)	0.051	0.121
glmnet-cox	0.090 (0.019)	0.728 (0.021)	0.113	0.002
xgboost-cox	0.083 (0.015)	0.730 (0.020)	2.632	0.003
nn-cox	-0.015 (0.048)	0.504 (0.037)	8.139	0.727
xgboost-aft	—	0.729 (0.021)	12.179	0.006
Lung cancer; death, n = 442, p = 24				
aorsf-cph	0.063 (0.031)	0.691 (0.019)	0.308	0.030
aorsf-fast	0.060 (0.033)	0.690 (0.019)	0.122	0.030
obliqueRSF-net	0.056 (0.018)	0.679 (0.021)	219.473	7.313
cif-extension	0.050 (0.018)	0.667 (0.019)	8.429	3.209
rsf-standard	0.050 (0.035)	0.673 (0.023)	1.081	0.072
cif-standard	0.050 (0.023)	0.667 (0.022)	0.318	0.924
ranger-extratrees	0.049 (0.016)	0.675 (0.019)	0.037	0.062
cif-rotate	0.047 (0.026)	0.664 (0.021)	16.753	2.820
aorsf-random	0.043 (0.021)	0.653 (0.024)	0.549	0.027
glmnet-cox	0.041 (0.024)	0.664 (0.034)	0.127	0.002
nn-cox	0.019 (0.038)	0.625 (0.062)	9.211	0.291
xgboost-cox	0.018 (0.019)	0.644 (0.027)	1.583	0.002
xgboost-aft	—	0.652 (0.026)	8.520	0.005
Monoclonal gammopathy; death, n = 1384, p = 8				
cif-rotate	0.159 (0.019)	0.744 (0.014)	15.330	4.515
aorsf-cph	0.158 (0.016)	0.743 (0.011)	1.176	0.092
aorsf-fast	0.157 (0.016)	0.743 (0.011)	0.407	0.091

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
cif-standard	0.151 (0.015)	0.738 (0.012)	1.512	6.113
rsf-standard	0.151 (0.017)	0.737 (0.011)	2.305	0.203
obliqueRSF-net	0.148 (0.009)	0.748 (0.011)	543.632	42.863
aorsf-random	0.148 (0.013)	0.738 (0.012)	1.747	0.086
cif-extension	0.143 (0.009)	0.747 (0.013)	10.794	4.507
glmnet-cox	0.137 (0.021)	0.726 (0.014)	0.146	0.002
xgboost-cox	0.122 (0.012)	0.733 (0.012)	4.230	0.003
ranger-extratrees	0.115 (0.005)	0.744 (0.012)	0.052	0.169
nn-cox	0.026 (0.051)	0.598 (0.100)	11.948	0.652
xgboost-aft	—	0.733 (0.013)	13.595	0.006
Monoclonal gammopathy; malignancy, n = 1384, p = 8				
glmnet-cox	0.015 (0.011)	0.651 (0.055)	0.129	0.002
obliqueRSF-net	0.012 (0.008)	0.649 (0.032)	143.443	22.157
aorsf-cph	0.010 (0.013)	0.644 (0.036)	0.594	0.041
aorsf-fast	0.010 (0.014)	0.641 (0.036)	0.190	0.041
ranger-extratrees	0.008 (0.006)	0.642 (0.030)	0.054	0.156
cif-extension	0.008 (0.010)	0.625 (0.028)	8.632	4.411
aorsf-random	0.007 (0.013)	0.636 (0.032)	0.532	0.040
cif-standard	0.006 (0.011)	0.628 (0.033)	1.490	5.778
xgboost-cox	0.005 (0.019)	0.639 (0.040)	1.686	0.003
nn-cox	-0.003 (0.005)	0.515 (0.056)	7.746	0.606
rsf-standard	-0.009 (0.018)	0.616 (0.036)	0.745	0.069
cif-rotate	-0.024 (0.023)	0.553 (0.035)	12.670	4.047
xgboost-aft	—	0.629 (0.039)	11.326	0.006

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
Movies released in 2015-2018; gross 1M USD, n = 551, p = 46				
cif-rotate	0.636 (0.024)	0.943 (0.007)	19.882	3.487
glmnet-cox	0.618 (0.034)	0.940 (0.009)	0.205	0.002
nn-cox	0.544 (0.055)	0.909 (0.020)	13.922	0.580
aorsf-cph	0.523 (0.024)	0.926 (0.011)	0.783	0.043
rsf-standard	0.519 (0.022)	0.922 (0.010)	1.503	0.103
aorsf-fast	0.516 (0.028)	0.922 (0.012)	0.227	0.043
xgboost-cox	0.512 (0.029)	0.932 (0.009)	13.524	0.004
cif-standard	0.472 (0.029)	0.902 (0.018)	0.354	1.715
cif-extension	0.454 (0.025)	0.920 (0.013)	9.152	3.724
ranger-extratrees	0.430 (0.025)	0.900 (0.019)	0.045	0.090
obliqueRSF-net	0.309 (0.020)	0.912 (0.017)	124.706	10.004
aorsf-random	0.303 (0.029)	0.851 (0.026)	0.950	0.042
xgboost-aft	—	0.927 (0.010)	35.466	0.007
NCCTG Lung Cancer; death, n = 228, p = 9				
ranger-extratrees	0.062 (0.028)	0.675 (0.033)	0.022	0.030
aorsf-random	0.061 (0.029)	0.676 (0.027)	0.324	0.015
aorsf-fast	0.061 (0.042)	0.672 (0.025)	0.066	0.017
aorsf-cph	0.059 (0.040)	0.671 (0.024)	0.153	0.016
obliqueRSF-net	0.056 (0.025)	0.678 (0.030)	88.165	3.793
cif-standard	0.055 (0.032)	0.670 (0.030)	0.128	0.254
cif-extension	0.051 (0.032)	0.664 (0.029)	3.845	1.378
glmnet-cox	0.033 (0.031)	0.638 (0.059)	0.097	0.002
rsf-standard	0.023 (0.039)	0.642 (0.025)	0.099	0.038

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
cif-rotate	0.017 (0.041)	0.632 (0.032)	4.906	1.275
xgboost-cox	0.012 (0.022)	0.648 (0.031)	1.076	0.002
nn-cox	-0.020 (0.019)	0.517 (0.110)	7.701	0.203
xgboost-aft	—	0.637 (0.034)	7.679	0.005
NKI 70 gene signature; death or metastasis, n = 144, p = 77				
aorsf-cph	0.124 (0.049)	0.802 (0.051)	0.074	0.014
aorsf-fast	0.121 (0.052)	0.802 (0.054)	0.049	0.015
cif-rotate	0.118 (0.059)	0.787 (0.049)	26.703	2.970
obliqueRSF-net	0.098 (0.049)	0.790 (0.062)	77.169	0.555
cif-extension	0.098 (0.055)	0.799 (0.061)	8.367	3.531
cif-standard	0.088 (0.051)	0.781 (0.065)	0.141	0.130
rsf-standard	0.087 (0.048)	0.755 (0.050)	0.066	0.025
ranger-extratrees	0.064 (0.044)	0.774 (0.054)	0.023	0.030
nn-cox	0.060 (0.065)	0.746 (0.059)	7.922	0.115
aorsf-random	0.051 (0.047)	0.733 (0.063)	0.150	0.015
glmnet-cox	0.049 (0.064)	0.726 (0.090)	0.271	0.002
xgboost-cox	-0.028 (0.029)	0.569 (0.094)	0.119	0.002
xgboost-aft	—	0.770 (0.056)	4.807	0.005
Non-alcohol fatty liver disease; death, n = 17549, p = 24				
aorsf-cph	0.213 (0.009)	0.869 (0.006)	17.803	1.370
aorsf-fast	0.212 (0.009)	0.869 (0.006)	4.902	1.336
rsf-standard	0.207 (0.009)	0.860 (0.005)	10.179	1.126
glmnet-cox	0.207 (0.011)	0.860 (0.005)	1.330	0.012
cif-standard	0.205 (0.007)	0.863 (0.006)	64.986	621.600

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
obliqueRSF-net	0.204 (0.008)	0.868 (0.006)	2703.887	9972.393
cif-rotate	0.190 (0.008)	0.865 (0.005)	259.239	60.313
ranger-extratrees	0.181 (0.007)	0.860 (0.005)	40.520	81.674
cif-extension	0.166 (0.003)	0.866 (0.006)	124.635	54.345
aorsf-random	0.141 (0.006)	0.839 (0.007)	9.973	1.490
xgboost-cox	0.022 (0.014)	0.876 (0.005)	9.315	0.017
nn-cox	0.000 (0.002)	0.557 (0.095)	19.415	103.251
xgboost-aft	—	0.875 (0.005)	31.562	0.014
Primary biliary cholangitis; death, n = 276, p = 19				
aorsf-fast	0.423 (0.035)	0.904 (0.021)	0.069	0.018
aorsf-cph	0.413 (0.034)	0.901 (0.022)	0.151	0.018
cif-rotate	0.405 (0.040)	0.899 (0.022)	9.295	2.069
rsf-standard	0.392 (0.034)	0.895 (0.023)	0.094	0.038
obliqueRSF-net	0.359 (0.030)	0.908 (0.022)	101.477	1.862
cif-standard	0.352 (0.034)	0.904 (0.025)	0.188	0.331
cif-extension	0.348 (0.033)	0.901 (0.023)	5.399	2.040
aorsf-random	0.344 (0.031)	0.891 (0.020)	0.277	0.019
glmnet-cox	0.342 (0.044)	0.886 (0.028)	0.117	0.002
ranger-extratrees	0.277 (0.027)	0.894 (0.027)	0.026	0.036
xgboost-cox	0.256 (0.103)	0.882 (0.026)	5.057	0.002
nn-cox	-0.024 (0.033)	0.556 (0.123)	8.423	0.195
xgboost-aft	—	0.883 (0.024)	9.373	0.006
Rotterdam tumor bank; death, n = 2982, p = 11				
aorsf-cph	0.163 (0.012)	0.759 (0.009)	2.494	0.205

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
aorsf-random	0.161 (0.011)	0.759 (0.010)	3.004	0.189
aorsf-fast	0.160 (0.012)	0.757 (0.009)	0.806	0.205
cif-standard	0.159 (0.010)	0.759 (0.009)	4.694	22.024
rsf-standard	0.159 (0.014)	0.756 (0.009)	2.995	0.391
obliqueRSF-net	0.156 (0.007)	0.759 (0.009)	931.931	64.305
cif-rotate	0.147 (0.011)	0.751 (0.011)	34.565	8.675
ranger-extratrees	0.139 (0.006)	0.749 (0.009)	3.211	2.477
xgboost-cox	0.130 (0.014)	0.753 (0.009)	4.472	0.004
cif-extension	0.129 (0.004)	0.751 (0.008)	22.084	8.110
glmnet-cox	0.118 (0.008)	0.731 (0.009)	0.247	0.003
nn-cox	-0.001 (0.001)	0.507 (0.049)	13.019	7.622
xgboost-aft	—	0.761 (0.009)	16.743	0.006
Rotterdam tumor bank; recurrence, n = 2982, p = 11				
aorsf-random	0.145 (0.011)	0.734 (0.009)	3.327	0.197
aorsf-cph	0.145 (0.012)	0.734 (0.009)	2.801	0.221
cif-standard	0.144 (0.011)	0.734 (0.009)	4.829	22.205
aorsf-fast	0.143 (0.011)	0.733 (0.009)	0.883	0.217
obliqueRSF-net	0.142 (0.008)	0.737 (0.009)	870.086	81.412
rsf-standard	0.139 (0.012)	0.731 (0.008)	3.113	0.947
ranger-extratrees	0.135 (0.007)	0.734 (0.009)	3.100	2.527
cif-rotate	0.129 (0.010)	0.725 (0.009)	36.405	8.349
cif-extension	0.119 (0.006)	0.731 (0.008)	22.537	8.390
glmnet-cox	0.117 (0.008)	0.727 (0.008)	0.227	0.004
xgboost-cox	0.113 (0.008)	0.729 (0.009)	4.123	0.004
nn-cox	-0.002 (0.002)	0.515 (0.029)	13.602	8.901

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
xgboost-aft	—	0.735 (0.009)	16.138	0.006
Serum free light chain; death, n = 7874, p = 10				
aorsf-fast	0.250 (0.014)	0.825 (0.008)	2.063	0.635
aorsf-cph	0.250 (0.013)	0.825 (0.008)	6.461	0.629
glmnet-cox	0.248 (0.012)	0.820 (0.007)	0.503	0.006
obliqueRSF-net	0.247 (0.011)	0.824 (0.007)	2219.216	1284.916
ranger-extratrees	0.243 (0.009)	0.820 (0.007)	11.176	10.433
cif-standard	0.243 (0.011)	0.818 (0.008)	19.000	116.158
rsf-standard	0.243 (0.013)	0.815 (0.008)	5.643	0.562
cif-rotate	0.228 (0.009)	0.819 (0.007)	63.456	20.143
aorsf-random	0.209 (0.011)	0.813 (0.008)	6.607	0.610
cif-extension	0.201 (0.005)	0.820 (0.008)	40.190	19.948
xgboost-cox	0.095 (0.038)	0.824 (0.007)	6.464	0.008
nn-cox	0.001 (0.003)	0.576 (0.111)	19.271	22.093
xgboost-aft	—	0.823 (0.008)	21.594	0.008
Systolic Heart Failure; death, n = 2231, p = 41				
glmnet-cox	0.113 (0.013)	0.745 (0.012)	0.268	0.003
cif-rotate	0.113 (0.013)	0.741 (0.011)	69.724	10.269
aorsf-cph	0.111 (0.014)	0.745 (0.012)	1.939	0.156
aorsf-fast	0.110 (0.015)	0.744 (0.012)	0.611	0.150
cif-standard	0.110 (0.011)	0.744 (0.011)	3.777	14.994
obliqueRSF-net	0.108 (0.009)	0.748 (0.012)	774.433	96.195
rsf-standard	0.105 (0.011)	0.735 (0.011)	2.783	0.272
aorsf-random	0.095 (0.008)	0.739 (0.012)	2.375	0.149

A.2: Index of prediction accuracy, time-dependent concordance statistic, and computational time required to fit and compute predictions for several learning algorithms across 21 risk prediction tasks. *(continued)*

	Scaled Brier	C-Statistic	Model fitting	Risk prediction
cif-extension	0.094 (0.006)	0.744 (0.012)	27.865	9.373
ranger-extratrees	0.091 (0.008)	0.738 (0.012)	3.445	1.214
xgboost-cox	0.091 (0.010)	0.744 (0.010)	4.688	0.004
nn-cox	0.076 (0.021)	0.710 (0.021)	14.766	4.725
xgboost-aft	—	0.741 (0.009)	14.633	0.006
VA lung cancer trial; death, n = 137, p = 8				
aorsf-cph	0.201 (0.052)	0.795 (0.034)	0.093	0.011
aorsf-fast	0.200 (0.050)	0.795 (0.034)	0.047	0.011
cif-rotate	0.198 (0.065)	0.789 (0.036)	4.005	1.004
rsf-standard	0.176 (0.048)	0.787 (0.037)	0.065	0.026
cif-extension	0.174 (0.048)	0.795 (0.034)	3.264	1.159
glmnet-cox	0.160 (0.036)	0.788 (0.037)	0.083	0.002
aorsf-random	0.151 (0.044)	0.777 (0.035)	0.205	0.012
cif-standard	0.128 (0.040)	0.770 (0.037)	0.093	0.119
obliqueRSF-net	0.114 (0.033)	0.799 (0.029)	53.069	0.734
ranger-extratrees	0.092 (0.033)	0.778 (0.038)	0.020	0.026
xgboost-cox	0.067 (0.076)	0.753 (0.045)	1.515	0.002
xgboost-aft	—	0.754 (0.047)	5.770	0.005
nn-cox	-0.030 (0.028)	0.521 (0.093)	7.791	0.118

A.3: Discrimination of relevant versus irrelevant variables for several techniques to estimate variable importance.

		accelerated oblique RSF			xgboost		RSF
Max correlation	No. observations	Negation	ANOVA	Permutation	SHAP	Gain	Permutation
Overall	Overall	75.9	73.9	73.2	69.6	64.6	67.7
Interactions							
Overall	Overall	57.8	57.4	58.0	54.6	49.2	56.9
30	500	54.3	54.1	54.8	48.2	42.7	54.9
30	1,000	56.9	55.7	58.1	53.1	48.0	56.3
30	2,500	61.9	58.9	64.1	61.5	60.7	60.0
15	500	53.1	53.5	52.8	47.1	41.1	54.1
15	1,000	55.9	55.4	56.3	52.2	45.8	55.4
15	2,500	61.0	58.6	63.0	61.0	58.9	59.9
0	500	52.5	53.9	52.4	44.5	40.7	53.6
0	1,000	57.2	58.6	55.8	53.1	42.8	56.1
0	2,500	67.6	68.2	64.4	71.0	62.2	62.1
Non-linear effects							
Overall	Overall	71.7	69.3	67.9	66.1	60.1	61.8
30	500	58.8	58.3	57.8	53.4	48.5	55.5
30	1,000	61.1	59.4	59.0	57.1	52.0	56.3
30	2,500	62.1	60.2	61.1	60.0	56.4	57.9
15	500	63.8	61.5	60.7	55.3	49.4	57.7

A.3: Discrimination of relevant versus irrelevant variables for several techniques to estimate variable importance. *(continued)*

Max correlation	No. observations	Negation	ANOVA	Permutation	SHAP	Gain	Permutation
15	1,000	67.5	65.1	64.6	62.5	56.0	59.8
15	2,500	70.2	67.2	69.1	66.8	62.3	62.3
0	500	75.5	72.3	68.5	60.1	55.8	61.1
0	1,000	88.3	83.9	78.0	81.5	68.6	67.6
0	2,500	98.4	96.3	91.8	97.7	91.6	78.3
Combination effects							
Overall	Overall	78.3	75.8	74.8	70.7	65.2	68.2
30	500	64.8	63.5	62.5	55.6	49.8	59.2
30	1,000	67.4	65.3	65.3	61.0	55.3	61.5
30	2,500	69.9	67.0	68.5	65.2	61.9	63.8
15	500	70.2	68.0	66.3	59.2	52.8	61.8
15	1,000	74.8	71.2	71.4	66.6	59.9	65.0
15	2,500	78.6	74.6	77.1	72.6	68.6	69.1
0	500	84.0	81.1	76.2	66.7	61.7	67.6
0	1,000	95.4	92.4	87.8	89.4	78.7	76.3
0	2,500	99.8	99.3	97.8	99.7	97.9	89.0
Main effects							
Overall	Overall	91.0	88.9	88.7	85.0	82.6	83.2
30	500	79.3	77.3	75.5	70.3	66.5	71.2
30	1,000	83.5	80.5	80.8	76.8	73.9	74.9

A.3: Discrimination of relevant versus irrelevant variables for several techniques to estimate variable importance. *(continued)*

Max correlation	No. observations	Negation	ANOVA	Permutation	SHAP	Gain	Permutation
30	2,500	86.5	83.5	85.1	81.7	80.4	79.3
15	500	86.3	83.3	81.8	75.7	71.3	75.3
15	1,000	91.3	88.1	88.5	84.6	81.3	81.1
15	2,500	94.5	91.6	93.7	90.2	89.0	86.5
0	500	97.8	96.3	94.0	86.5	83.4	85.9
0	1,000	100.0	99.7	99.4	99.4	98.0	95.2
0	2,500	100.0	100.0	100.0	100.0	100.0	99.8

References

Allen, A. M., T. M. Therneau, J. J. Larson, A. Coward, V. K. Somers, and P. S. Kamath (2018). Nonalcoholic fatty liver disease incidence and impact on metabolic burden and death: a 20 year-community study. *Hepatology 67*(5), 1726–1736.

ARIC Investigators (1989). The atherosclerosis risk in communit (aric) study: design and objectives. *American journal of epidemiology 129*(4), 687–702.

Benavoli, A., G. Corani, J. Demšar, and M. Zaffalon (2017). Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *The Journal of Machine Learning Research 18*(1), 2653–2688.

Bild, D. E., D. A. Bluemke, G. L. Burke, R. Detrano, A. V. Diez Roux, A. R. Folsom, P. Greenland, D. R. JacobsJr, R. Kronmal, K. Liu, et al. (2002). Multi-ethnic study of atherosclerosis: objectives and design. *American journal of epidemiology 156*(9), 871–881.

Blanche, P., J.-F. Dartigues, and H. Jacqmin-Gadda (2013). Estimating and comparing time-dependent areas under receiver operating characteristic curves for censored event times with competing risks. *Statistics in medicine 32*(30), 5381–5397.

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (2017). *Classification and regression trees*. Routledge.

Brilleman, S. (2018). *simsurv: Simulate Survival Data*. R package version 0.2.2, available at `https://CRAN.R-project.org/package=simsurv`.

Brilleman, S. L., R. Wolfe, M. Moreno-Betancur, and M. J. Crowther (2020). Simulating survival data using the simsurv R package. *Journal of Statistical Software 97*(3), 1–27.

Casalicchio, G., J. Bossek, M. Lang, D. Kirchhoff, P. Kerschke, B. Hofner, H. Seibold, J. Vanschoren, and B. Bischl (2017). OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, 1–15.

Chen, T., T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, Y. Li, and J. Yuan (2022). *xgboost: Extreme Gradient Boosting*. R package version 1.5.2.1.

Cui, Y., R. Zhu, M. Zhou, and M. Kosorok (2017). Consistency of survival tree and forest models: splitting bias and correction. *arXiv preprint arXiv:1707.09631*.

Desmedt, C., A. Di Leo, E. de Azambuja, D. Larsimont, B. Haibe-Kains, J. Selleslags, S. Delaloge, C. Duhem, J.-P. Kains, B. Carly, et al. (2011). Multifactorial approach to predicting resistance to anthracyclines. *Journal of Clinical Oncology 29*(12), 1578–1586.

Director's Challenge Consortium for the Molecular Classification of Lung Adenocarcinoma (2008). Gene expression–based survival prediction in lung adenocarcinoma: a multi-site, blinded validation study. *Nature medicine 14*(8), 822–827.

Dispenzieri, A., J. A. Katzmann, R. A. Kyle, D. R. Larson, T. M. Therneau, C. L. Colby, R. J. Clark, G. P. Mead, S. Kumar, L. J. Melton III, et al. (2012). Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population. In *Mayo Clinic Proceedings*, Volume 87, pp. 517–523. Elsevier.

Felker, G. M., K. J. Anstrom, K. F. Adams, J. A. Ezekowitz, M. Fiuzat, N. Houston-Miller, J. L. Januzzi, D. B. Mark, I. L. Piña, G. Passmore, et al. (2017). Effect of natriuretic peptide–guided therapy on hospitalization or cardiovascular mortality in high-risk patients with heart failure and reduced ejection fraction: a randomized clinical trial. *Jama 318*(8), 713–720.

Goodrich, B., J. Gabry, I. Ali, and S. Brilleman (2022). rstanarm: Bayesian applied regression modeling via Stan. R package version 2.21.3.

Graf, E., C. Schmoor, W. Sauerbrei, and M. Schumacher (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine 18*(17-18), 2529–2545.

Harrell, F. E., R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati (1982, 05). Evaluating the Yield of Medical Tests. *JAMA 247*(18), 2543–2546.

Hatzis, C., L. Pusztai, V. Valero, D. J. Booser, L. Esserman, A. Lluch, T. Vidaurre, F. Holmes, E. Souchon, H. Wang, et al. (2011). A genomic predictor of response and survival following taxane-anthracycline chemotherapy for invasive breast cancer. *Jama 305*(18), 1873–1881.

Heath, D., S. Kasif, and S. Salzberg (1993). Induction of oblique decision trees. In *IJCAI*, Volume 1993, pp. 1002–1007. Citeseer.

Heller, G. (2012). A measure of explained risk in the proportional hazards model. *Biostatistics 13*(2), 315–325.

Hooker, G., L. Mentch, and S. Zhou (2021). Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing 31*(6), 1–16.

Horst, A. M., A. P. Hill, and K. B. Gorman (2020). *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*. R package version 0.1.0.

Hosmer, D. W. and S. Lemeshow (2002). *Applied survival analysis: regression modelling of time to event data*. Wiley.

Hothorn, T. (2022). *TH.data: TH's Data Archive*. R package version 1.1-1.

Hothorn, T., K. Hornik, C. Strobl, and A. Zeileis (2010). Party: a laboratory for recursive partytioning.

Hothorn, T., K. Hornik, and A. Zeileis (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics 15*(3), 651–674.

Hsich, E., E. Z. Gorodeski, E. H. Blackstone, H. Ishwaran, and M. S. Lauer (2011). Identifying important risk factors for survival in patient with systolic heart failure using random survival forests. *Circulation: Cardiovascular Quality and Outcomes 4*(1), 39–45.

Hvitfeldt, E. and H. Frick. *censored: 'parsnip' Engines for Survival Models*. R package version 0.1.0.9000.

Ishwaran, H. and U. Kogalur (2019). *Random Forests for Survival, Regression, and Classification (RF-SRC)*. R package version 2.8.0, available at `https://cran.r-project.org/package=randomForestSRC`.

Ishwaran, H. and U. B. Kogalur (2010). Consistency of random survival forests. *Statistics & probability letters 80*(13-14), 1056–1064.

Ishwaran, H., U. B. Kogalur, E. H. Blackstone, and M. S. Lauer (2008). Random survival forests. *The Annals of Applied Statistics*, 841–860.

Ishwaran, H. and M. Lu (2019). Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival. *Statistics in medicine 38*(4), 558–582.

Jaeger, B. C. (2022). *aorsf: Accelerated Oblique Random Survival Forests*. R package version 1.0.0.

Jaeger, B. C., D. L. Long, D. M. Long, M. Sims, J. M. Szychowski, Y.-I. Min, L. A. Mcclure, G. Howard, and N. Simon (2019). Oblique random survival forests. *The Annals of Applied Statistics 13*(3), 1847–1883.

Jaeger, B. C., S. Welden, K. Lenoir, and N. M. Pajewski (2022). aorsf: An R package for supervised learning using the oblique random survival forest. *Journal of Open Source Software 7*(77), 4705.

Kalbfleisch, J. D. and R. L. Prentice (2011). *The statistical analysis of failure time data*. John Wiley & Sons.

Kattan, M. W. and T. A. Gerds (2018). The index of prediction accuracy: an intuitive measure useful for evaluating risk prediction models. *Diagnostic and prognostic research 2*(1), 1–7.

Katuwal, R., P. N. Suganthan, and L. Zhang (2020). Heterogeneous oblique random forest. *Pattern Recognition 99*, 107078.

Kuhn, M. and H. Wickham (2020). *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles.*

Kyle, R. A., T. M. Therneau, S. V. Rajkumar, D. R. Larson, M. F. Plevak, J. R. Offord, A. Dispenzieri, J. A. Katzmann, and L. J. Melton III (2006). Prevalence of monoclonal gammopathy of undetermined significance. *New England Journal of Medicine 354*(13), 1362–1369.

Kyle, R. A., T. M. Therneau, S. V. Rajkumar, J. R. Offord, D. R. Larson, M. F. Plevak, and L. J. Melton III (2002). A long-term study of prognosis in monoclonal gammopathy of undetermined significance. *New England Journal of Medicine 346*(8), 564–569.

Landau, W. M. (2021). The targets r package: a dynamic make-like function-oriented pipeline toolkit for reproducibility and high-performance computing. *Journal of Open Source Software 6*(57), 2959.

Li, H., D. Han, Y. Hou, H. Chen, and Z. Chen (2015). Statistical inference methods for two crossing survival curves: a comparison of methods. *PLoS One 10*(1), e0116774.

Loprinzi, C. L., J. A. Laurie, H. S. Wieand, J. E. Krook, P. J. Novotny, J. W. Kugler, J. Bartel, M. Law, M. Bateman, and N. E. Klatt (1994). Prospective evaluation of prognostic variables from patient-completed questionnaires. north central cancer treatment group. *Journal of Clinical Oncology 12*(3), 601–607.

Lundberg, S. and S.-I. Lee (2017). A unified approach to interpreting model predictions.

Lundberg, S. M., G. G. Erion, and S.-I. Lee (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Menze, B. H., B. M. Kelm, D. N. Splitthoff, U. Koethe, and F. A. Hamprecht (2011). On oblique random forests. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 453–469. Springer.

Moertel, C. G., T. R. Fleming, J. S. Macdonald, D. G. Haller, J. A. Laurie, C. M. Tangen, J. S. Ungerleider, W. A. Emerson, D. C. Tormey, J. H. Glick, et al. (1995). Fluorouracil plus levamisole as effective adjuvant therapy after resection of stage iii colon carcinoma: a final report. *Annals of internal medicine 122*(5), 321–326.

Moons, K. G., A. P. Kengne, M. Woodward, P. Royston, Y. Vergouwe, D. G. Altman, and D. E. Grobbee (2012). Risk prediction models: I. development, internal validation, and assessing the incremental value of a new (bio) marker. *Heart 98*(9), 683–690.

Moss, L., D. Corsar, M. Shaw, I. Piper, and C. Hawthorne (2022). Demystifying the black box: The importance of interpretability of predictive models in neurocritical care. *Neurocritical care*, 1–7.

Murthy, S. K., S. Kasif, and S. Salzberg (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research 2*, 1–32.

Pintilie, M. (2006). *Competing risks: a practical perspective*. John Wiley & Sons.

Poona, N., A. Van Niekerk, and R. Ismail (2016). Investigating the utility of oblique tree-based ensembles for the classification of hyperspectral data. *Sensors 16*(11), 1918.

Qiu, X., L. Zhang, P. N. Suganthan, and G. A. Amaratunga (2017). Oblique random forest ensemble via least square estimation for time series forecasting. *Information Sciences 420*, 249–262.

Rainforth, T. and F. Wood (2015). Canonical correlation forests. *arXiv preprint arXiv:1507.05444*.

Royston, P. and D. G. Altman (2013). External validation of a cox prognostic model: principles and methods. *BMC medical research methodology 13*(1), 1–15.

Schumacher, M. (1994). Rauschecker for the german breast cancer study group, randomized 2 by 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. *Journal of Clinical Oncology 12*, 2086–2093.

Sonabend, R., F. J. Király, A. Bender, B. Bischl, and M. Lang (2021, 02). mlr3proba: An r package for machine learning in survival analysis. *Bioinformatics*.

SPRINT Research Group (2015). A randomized trial of intensive versus standard blood-pressure control. *New England Journal of Medicine 373*(22), 2103–2116.

Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics 8*(1), 25.

Taylor Jr, H. A., J. G. Wilson, D. W. Jones, D. F. Sarpong, A. Srinivasan, R. J. Garrison, C. Nelson, and S. B. Wyatt (2005). Toward resolution of cardiovascular health disparities in african americans: Design and methods of the jackson heart study. *Ethn Dis 15*(4 Suppl 6), S6–4.

Ternès, N., F. Rotolo, G. Heinze, and S. Michiels (2017). Identification of biomarker-by-treatment interactions in randomized clinical trials with survival outcomes and high-dimensional spaces. *Biometrical Journal 59*(4), 685–701.

Ternes, N., F. Rotolo, and S. Michiels (2018). *biospear: Biomarker Selection in Penalized Regression Models*. R package version 1.0.2.

Therneau, T. (2022a, April). Survival package source code documentation. original-date: 2016-04-28.

Therneau, T. M. (2022b). *A Package for Survival Analysis in R*. R package version 3.3-1.

Therneau, T. M. and P. M. Grambsch (2000). *Modeling Survival Data: Extending the Cox Model*. Springer.

Tomita, T. M., J. Browne, C. Shen, J. Chung, J. L. Patsolic, B. Falk, C. E. Priebe, J. Yim, R. Burns, M. Maggioni, et al. (2020). Sparse projection oblique randomer forests. *Journal of machine learning research 21*(104).

Uno, H., T. Cai, M. J. Pencina, R. B. D'Agostino, and L.-J. Wei (2011). On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine 30*(10), 1105–1117.

Van De Vijver, M. J., Y. D. He, L. J. Van't Veer, H. Dai, A. A. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, et al. (2002). A gene-expression signature as a predictor of survival in breast cancer. *New England Journal of Medicine 347*(25), 1999–2009.

Wang, H. and G. Li (2017). A selective review on random survival forests for high dimensional data. *Quantitative bio-science 36*(2), 85.

Wang, H. and L. Zhou (2017). Random survival forest with space extensions for censored data. *Artificial intelligence in medicine 79*, 52–61.

Wright, M. N. and A. Ziegler (2017). ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software 77*(1), 1–17.

Zhang, L. and P. N. Suganthan (2014). Oblique decision tree ensemble via multisurface proximal support vector machine. *IEEE transactions on cybernetics 45*(10), 2165–2176.

Zhou, L., H. Wang, and Q. Xu (2016). Random rotation survival forest for high dimensional censored data. *SpringerPlus 5*(1), 1–10.

Zhu, R. (2013). *Tree-based Methods for Survival Analysis and High-dimensional Data*. Ph. D. thesis, The University of North Carolina at Chapel Hill.

Zhu, R., D. Zeng, and M. R. Kosorok (2015). Reinforcement learning trees. *Journal of the American Statistical Association 110*(512), 1770–1784.